1
+ 1
———
10

数制浅说

上海人民出版社

数 制 浅 说

上海师范大学数学系计算数学组

上 海 人 民 出 版 社

内 容 提 要

本书通俗地介绍了几种常用的数制以及二进制与电子计算机的一些关系。

全书共分三个部分：第一部分从历史上叙述了数和数制概念的产生情况；第二部分较详细地讨论了十进制、二进制、八进制、十六进制以及它们之间相互转换的方法；第三部分介绍了二进制与电子计算机的一些关系。

本书可供中学生课外阅读，也可供需要了解数制知识的读者参考。

数 制 浅 说

上海师范大学数学系计算数学组

上海人民出版社出版

（上海绍兴路5号）

新华书店上海发行所发行　上海市印刷十二厂印刷

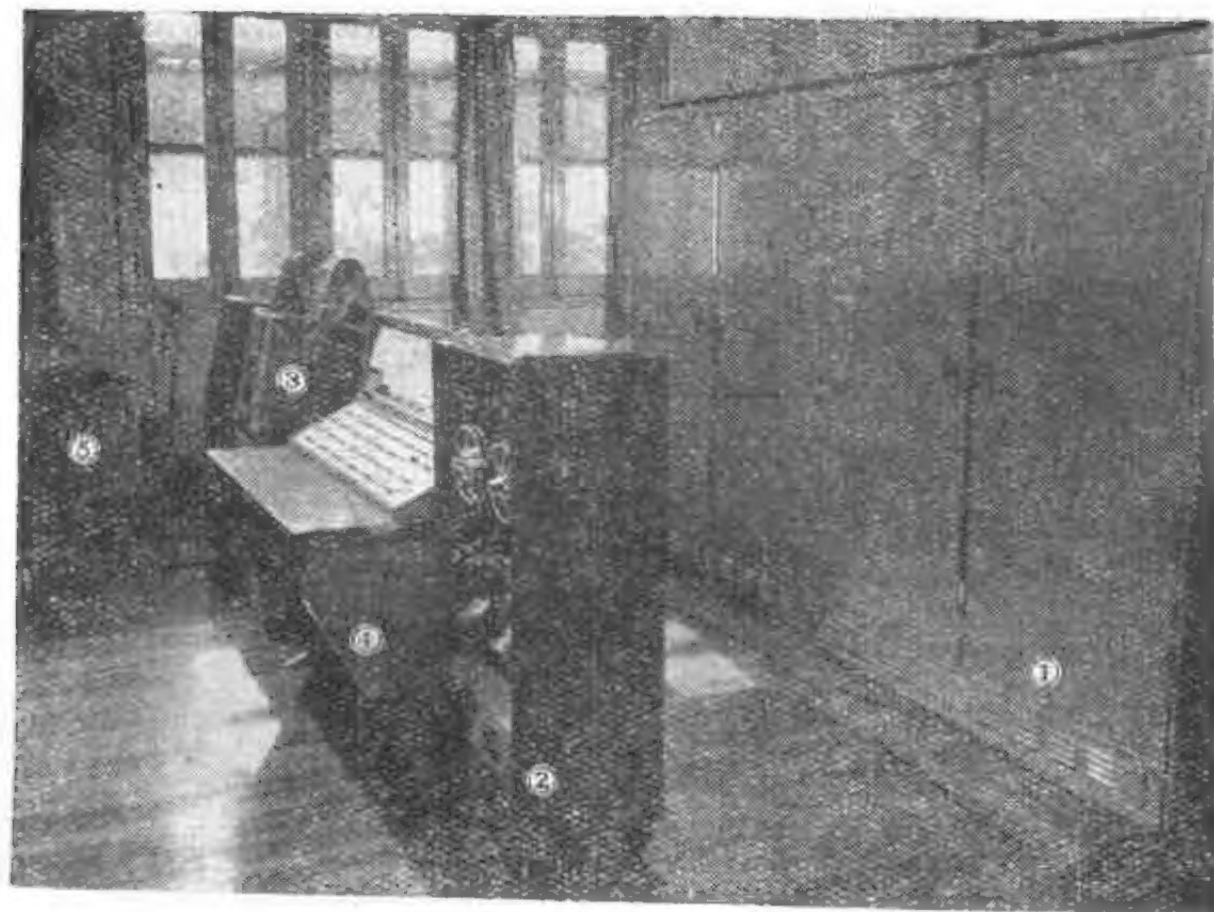
开本 787×1092 1/32　印张 2.5　字数 53,000

1974年6月第1版　1974年6月第1次印刷

印数 1-80,000

统一书号：13171·92　　定价：0.17元

在宽敞、明亮、整洁的电子计算机室中的
国产集成电路数字电子计算机



① 运算器、内存储器、控制器；
② 光电输入机；
③ 输出机；
④ 人工控制台；
⑤ 磁鼓（外存储器）

目录

前　言

你参观过电子计算机吗？当你走进宽敞、明亮、整洁的电子计算机室时，你一定会看到：一盘盘穿有不同小孔的纸带在计算人员的操作下，飞快地输入到机器中去；机器立刻迅速、准确地完成了成千上万次的运算；然后，在输出机上清晰地打印出一行行所需要的计算结果。

这时，你看着一台台运转着的国产精密电子计算机，一定会为我国工人阶级在毛主席的无产阶级革命路线指引下，坚持"**独立自主、自力更生**"的方针所取得的伟大胜利而感到无比自豪。同时你还会想到，生产实际问题中的数字和计算公式是怎样告诉电子计算机的？它与这一盘盘穿了孔的纸带有什么关系呢？还有，我们平时所接触到的数字一般都是十进数，那么在电子计算机中是否也是用十进数进行计算呢？如果有人告诉你机器并不是用十进数进行运算的，而是用二进数进行运算的，那么你也许会问：什么叫二进数？二进数与十进数之间有什么关系？还有哪些常用的数制等等。

在这本书中，我们想通过介绍一些常用的数制和它们之间的相互转换关系，以及二进数与电子计算机之间的一些关系，来解答以上提出的一些问题。

一、数与数制

1. 数的概念的产生

在日常生活中，我们经常用到数及数的运算．在阶级斗争、生产斗争和科学实验三大革命斗争中也要注意事物的数量方面，才能把事情办好．目前，就一般人来说，即使是入学不久的儿童，都能认识一些简单的数，并且能进行简单的计算；然而数的概念的产生却是人类长期生产实践的结果，它经历了漫长的历史发展过程．

毛主席教导我们说：“**人类的生产活动是最基本的实践活动，是决定其他一切活动的东西。**”恩格斯指出：“**数和形的概念不是从其他任何地方，而是从现实世界中得来的。**”（《反杜林论》第35页）人类在长期的生产实践活动中，在分析和概括大量实践经验的基础上，产生了自然数1、2、3、4、…的概念．在原始社会中，人们从事狩猎和捕鱼活动，起初，他们只能判断捕获物的多少，也就是已有了数的感觉．随着人类社会生产活动的进一步发展，人们接触到各种物体的集合（我们将某种物体的全体称为由这种物体所组成的集合），于是逐渐产生了与具体物体的集合相联系的数；用于不同种类的物体，数有不同的名称，一些是用来计算人的，另一些是用来计算船只的等等；这些“有名数”是分别属于一定种类的物体的，这里还不是抽象的“数”．恩格斯指出：“**为了计数，不仅要有可以计数的对象，而且还要有一种在考察对象时撇开对象的其他一切特性而仅仅顾到数目的能力，而这种能力是长期的以经验为**

依据的历史发展的结果。”(«反杜林论»第35页)人们对不少物体的集合进行比较，于是能够进一步由建立各个物体集合的元素(集合中每一个物体称为该集合的元素)之间的对应来判断集合所含物体数是否一样．例如，拿参加捕获的人与他们的捕获物进行比较，判断他们的个数多少．我国上古时代有“结绳而治”的说法，就是在绳子上挽成结子，以结子的个数来表示事物的件数．在此，人们认识到物体的数目是物体集合的一种性质，为了明显地分辨这种共同性质，就产生了抽象的“数”的概念，于是就有了自然数1、2、3、4、…．分数是在度量中产生的．人们在丈量土地和测量容积中发现：一般说来，所选用的单位不能在被测量的量上置放整数次，所以在测量时不能满足于简单计算有多少个单位．这时必须把单位加以分划，以便利用单位的一部分来更准确地表示量，这样就产生了分数．至于无理数刻划了不可通约的量，正负数表示具有相反意义的量等等，在此就不多述了．

2．数制的概念

随着生产工具的改进和生产力的逐步提高，人们需要对较多量的事物进行计数，因而需要用到较大的数．对于这样大量的数，人们怎样给出它们的名称和怎样书写呢？为了解决这个问题，就产生了进位的方法和各种不同的数制．

在古代社会，人们已经使用了十进制计数法．在殷代遗留下来的甲骨文字中，自然数的记法已经毫无例外地用着十进制．我们现在习惯使用的也是十进制．由于人们生来有十个指头，古代人类就用这十个指头计数，于是就创造了一、二、三、四、五、六、七、八、九、十这十个数字．当他们对较大量的物体进行计数时，先将十个物体放在一堆，算作一个单

位；将所有的物体分成这样一堆一堆的，得到若干堆，可能还剩下不足十个物体的零头；在堆数超过十时，又将每十堆算作一个单位，依次继续进行下去．在这样的计数过程中，就有了“百”、“千”、“万”等等．这就是十进制计数法．

与十进制计数法相配合的是十进制记数法．人们用阿拉伯数码 1、2、3、4、5、6、7、8、9 表示前九个自然数，数码 0 在书写数字时起着“填空作用”．满了十个，就在第二位上写上 1，在第一位上写上 0，即写成 10；同样，十个十（即百）就在第三位上写上 1，在第二位和第一位上写上 0，即写成 100 等等．并排写着的数码 374 表示三个“百”、七个“十”与四个“一”的和．一般来说，并排写着的几个数码，它的最右面一个数码表示第一位（个位）上的单位“一”的个数，紧靠着它的左面的一个数码表示第二位（十位）上的单位“十”的个数，再左面的一个数码表示第三位（百位）上的单位“百”的个数等等．在十进制记数法里，首先我们有了十个数码：0、1、2、3、4、5、6、7、8、9；再注意到各个数位间的关系：第一位（最右面一位）上的一个单位是一；任何两个紧靠着的数位，左面一位上的一个单位是右面一位上的一个单位的十倍．利用十个数码和位置法则——并排写着的数码，不仅是数码本身，而且它所在的位置也具有意义——就可写出任意大的整数了．

回顾一下历史，人们也并不是一开始就使用阿拉伯数字和掌握用进位法来记数的．我国古代采用十进制记数法，很早就有了一、二、三、四、五、六、七、八、九、十、百、千、万等方块字，用它们来记数．到唐代，有了代表一、二、三、四、五、六、七、八、九这九个数字的数码，写法如下：

	一	二	三	四	五	六	七	八	九
纵式	𝍠	𝍡	𝍢	𝍣	𝍤	𝍥	𝍦	𝍧	𝍨

横式　𝍩　𝍪　𝍫　𝍬　𝍭　𝍮　𝍯　𝍰　𝍱

到明、清时代，又演化成：

	一	二	三	四	五	六	七	八	九	零
纵式	〡	〢	〣	〤	〥	〦	〧	〨	〩	〇
横式	一	二	三	〤	〥	〦	〧	〨	〩	〇

但是，我国古代的数字计算是用筹算，到十四世纪以后又发明了珠算，它们分别是用算筹和算珠进行计算的，而不用笔算．因此，当时虽有数码，但并不用它进行计算．

十三世纪以前，欧洲各国盛行罗马数码．用七个符号

I　V　X　L　C　D　M

分别代表数

一　五　十　五十　百　五百　千

将这些符号按照下面的规则组合起来，就能够表示任何整数．

(1) 用“重复书写几次”的方法，表示某一符号代表的数出现几次．

例如符号 I 表示数一，于是二和三就写成：

II、III．

符号 X 表示数十，于是二十、三十就写成：

XX、XXX．

用同样的方法可以写出二百、三百、二千、三千．

(2) 用“右加左减”的法则可写出另一些数．若在一个数码的右边附着一个较小的数码，就表示大数加小数的和．

例如，五、六、七、八分别写成：

V、VI、VII、VIII．

而十、十一、十二，则分别写成：

X、XI、XII．

若在一个数码的左边附着一个较小的数码，就表示大数

减去小数的差。

例如，四、九、四十、九十、九十九等分别写成：

IV、IX、XL、XC、IC。

(3) 用"加一横线"的方法表示千以上的数。若在某数上加一横线，就表示这个数的一千倍。

例如，一万九千七百二十六写成：

$\overline{XIX}$DCCXXVI；

也可写成：

XIX$_{m}$DCCXXVI。

在这里，写得小一些的字母 m 表示千。

在罗马数字记数法中，虽然只有七个不同的符号，但是对于不太大的数目就要写得很长，并且用于计算也很不方便，因此现在已很少使用，在有的钟表和书上还可见到罗马数字。

我们现在所用的"阿拉伯"数字，最先是在印度发明使用的，以后传入阿拉伯。在十二世纪再从阿拉伯转输入欧洲各国，而被称为"阿拉伯"数字。

以上我们讲到十进制计数法和十进制记数法。在日常生活中或在通常情况下，我们采用十进制记数法进行计算很为方便。但是，除了十进制以外，还有其他的进位制。在商业上用到十二进制。例如，十二支铅笔叫做一"打"；十二"打"叫做一"箩"。特别，时间和角度的单位是采用六十进制。古代人类由于生产劳动的需要，从事天文和历法的研究。他们观测地球的自转，研究昼夜的变化，自转的角度和经历的时间密切相关。为了提高观测结果的精确程度，以适应制定历法的需要，人们将计算时间的单位"小时"和计算角度的单位"度"进一步划分：以一小时作为六十分钟，一分钟作为六十秒钟；相应地以一度作为六十分，一分作为六十秒，这就是沿用至今的

六十进制．虽然我们在日常生活中用得最多的是十进制，但六十进制在某些方面也有方便的地方．例如，在不足一小时时，$\frac{1}{2}$小时＝30分，$\frac{1}{3}$小时＝20分，$\frac{1}{4}$小时＝15分，$\frac{1}{5}$小时＝12分，$\frac{1}{6}$小时＝10分等等，都得到较小的单位“分”的整数倍．而在十进制里，$\frac{1}{3}$则是无限小数了．

还有一种很有用的进位制，就是二进制．在二进制里，只需要两个数码：0与1．数“一”是个位上的一个单位，用数码1表示；数“二”就构成第二位上的一个单位，记作10；数“三”由第二位上的一个单位“二”和个位上的一个单位“一”所组成，所以写成11；数“四”包含两个“二”(即二的平方)，因此需要进到第三位上了，写作100；至于数“八”它是二的立方，所以它的写法应和十进制的“千”一样，写成1000．在大多数电子计算机里都是对二进数进行运算的，利用二进数进行运算时，与它密切关联的还要用到八进数、十六进数等等．

随着我国社会主义革命和社会主义建设的不断深入发展，各条战线上形势大好，工业和近代科学技术正在蓬勃发展．在毛主席的无产阶级革命路线指引下，我国第一台每秒钟运算一百万次的集成电路电子计算机，已经试制成功，这标志着我国电子计算技术又前进了一步．由于电子计算机的使用日益广泛，人们对于了解有关二进制记数法方面的知识，就有了较普遍的要求．

二、不同数制间的相互转换

在前一节里，我们看到在通常情况下，人们采用十进制记数法；但在解决不同的问题时，也有采用其他进位制的，而且更为方便．在这一节里，我们就要介绍四种常用的数制——十进制、二进制、八进制和十六进制，并且给出它们之间相互转换的方法．

1．数字的表达原理

“一切客观事物本来是互相联系的和具有内部规律的”．各种不同的进位制都是用数字刻划客观事物的数量，而且基本方法是一样的．**“我们看事情必须要看它的实质，而把它的现象只看作入门的向导，一进了门就要抓住它的实质，这才是可靠的科学的分析方法。”**为了了解在各种不同数制下数字表达的原理，我们先看十进制记数法，掌握了它的实质，对于其他的进位制就容易理解了．

1.1 十 进 制

人们根据“逢十进一”的原则进行计数，因而有了十进制计数法，与它相配合的是十进制记数法．

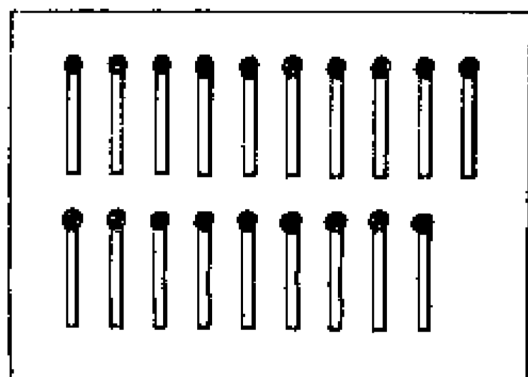


图 1

我们数一数图 1 中有多少根火柴．试想：怎样用数字表示出火柴的数量？

在十进制记数法中，有十个不同的数码：0、1、2、3、4、

5、6、7、8、9；根据“逢十进一”以及前节中所讲到的位置法则，就能用数字表达事物的数量，这就是十进制中数字的表达原理．我们称“十”为十进制记数法的“基数”．

我们数图 1 中的火柴，把十根火柴作为一堆时，得到一堆和九根零头；于是将数得的结果记作 19．在此

$$19=1\times 10+9\times 1.$$

除此而外，上式还可以进一步写作

$$\begin{aligned}19&=1\times 10+9\times 1\\&=1\times 10^1+9\times 10^0.\end{aligned}$$

注意，零次幂的意义：$10^0=1$．

同样，在十进制中，数

$$\begin{aligned}256&=2\times 100+5\times 10+6\times 1\\&=2\times 10^2+5\times 10^1+6\times 10^0.\end{aligned}$$

我们从上面的表达式看出：在十进制记数法中的数（称为十进数）容易写成基数“十”的各次乘幂的和的形式．此十进数的各个数位上的数码（上例中的 2、5、6）也就是所作的和式中各次乘幂的系数．第一位（自右至左）上的数码(6)是和式中“十”的零次幂的系数，第二位上的数码(5)是“十”的一次幂的系数，第三位上的数码(2)是“十”的二次幂的系数等等．根据这个法则，十进数

$$\begin{aligned}347012=&3\times 10^5+4\times 10^4+7\times 10^3\\&+0\times 10^2+1\times 10^1+2\times 10^0.\end{aligned}$$

正如数位上出现的“0”不可略去一样，表达式中系数是“0”的项同样要写出，这样一个数的各个数位上的数码才能与右面表达式中各项系数对应起来．

对于十进小数，我们注意小数点后第一位（自左至右）上的一个单位是个位上一个单位的十分之一，小数点后第二位

上一个单位是小数点后第一位上一个单位的十分之一(即为个位上一个单位的百分之一)等等．于是，十进小数也可写作“十”的各次乘幂的和的形式，这里出现了负整数指数幂．

例如：

$$0.867=8\times\frac{1}{10}+6\times\frac{1}{100}+7\times\frac{1}{1000}$$
$$=8\times10^{-1}+6\times10^{-2}+7\times10^{-3}.$$

注意，负数指数幂的意义：

$$10^{-1}=\frac{1}{10},\quad 10^{-2}=\frac{1}{10^2},\quad 10^{-3}=\frac{1}{10^3}\text{ 等等}.$$

同样 $190.036=1\times10^2+9\times10^1+0\times10^0$
$+0\times10^{-1}+3\times10^{-2}+6\times10^{-3}.$

从上面的例子中，我们看到十进数与将它写成关于基数“十”的各次乘幂的和式之间，有着明显的联系，这也是十进制记数法的一个根本性质．

“矛盾的普遍性即寓于矛盾的特殊性之中。”我们明了了十进制记数法的实质，就不难明白其他的进位制了．

1.2 二 进 制

当我们根据“逢二进一”的法则进行计数时，就得到二进制计数法．试用二进制计数法来数图1中的火柴的根数．怎样用二进制记数法将这个数写出来呢？

在二进制里有两个不同的数码0与1，根据“逢二进一”与相应的位置法则记数．在这里，第一位(自右至左)上一个单位是“一”；第二位上一个单位是“一”的两倍，即“二”；第三位上一个单位又是“二”的两倍，即“四”；这样下去，第四位上一个单位是“八”；第五位上一个单位是“十六”等等．我们称二进制的基数是“二”．

我们来数火柴，先将每二根并成一堆，得到九堆，还剩下一根(图 2)；由于堆数超过了“二”，又将每两堆并成一个较大的堆，得到四堆(每堆由四根火柴组成)，又剩下一个由两根火柴组成的堆(图 3)；再将这四堆火柴每两堆并成一堆，得到两个更大的堆(每堆由八根组成，见图 4)；最后再将这两堆并成一堆．这样以来，我们就得到了一个由“十六”根火柴组成的堆，一个由“二”根火柴组成的堆，另外还有一根(图 5)．注意这里没有由“八”根火柴及由四根火柴组成的堆．

火柴根数用二进数表示就是

$$10011.$$

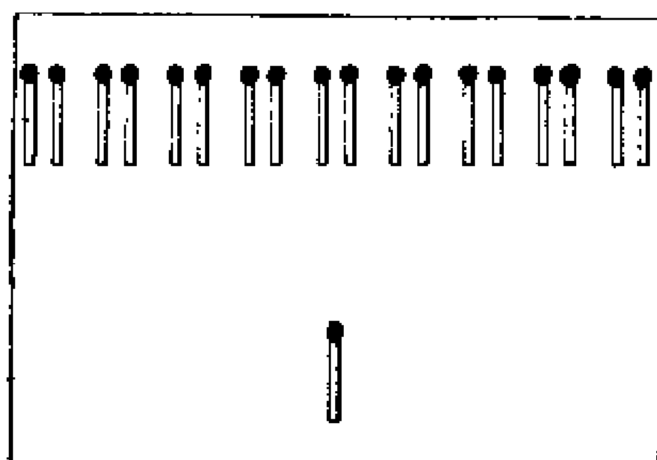


图 2

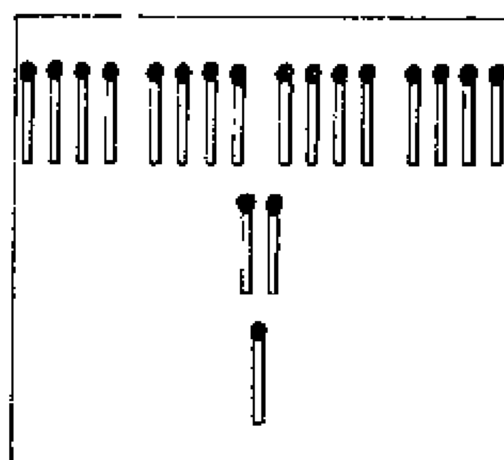


图 3

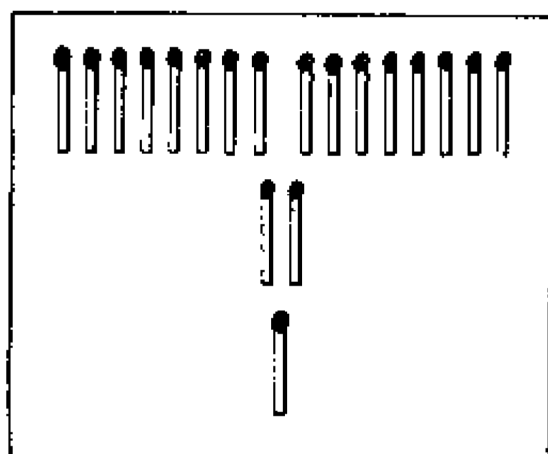


图 4

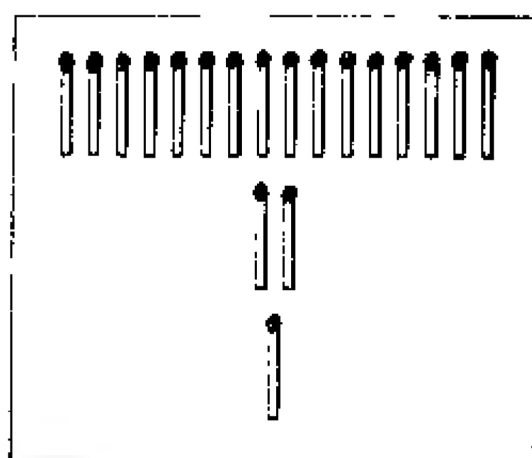


图 5

根据位置法则，不难写出下列十进数与二进数的对照表：

十进数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	…
二进数	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	…

例如，二进数 101 是"四"与"一"的和，即十进数 5；二进数 1101 是"八"、"四"与"一"的和，即十进数 13.

上面所讲的是二进整数，第一位（自右至左）上一个单位是"一"．对于相邻的两个数位来说，左面一位上一个单位是右面一位上一个单位的两倍；反过来，右面一位上一个单位是左面一位上一个单位的二分之一．对于二进小数，各个数位间有同样的关系．小数点后第一位（自左至右）上一个单位是它的左面一位上——小数点前一位上的一个单位的二分之一，即"一"的二分之一；小数点后第二位上一个单位是"二分之一"的二分之一，即"四分之一"；小数点后第三位上一个单位是"八分之一"，第四位上一个单位是"十六分之一"等等．从而，即有如下的对应关系：

二进数	0.1	0.01	0.001	0.0001	0.00001	…
十进数	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	…

由二进制的位置法则，任意一个二进数容易写成基数"二"的乘幂的和的形式．下面我们用（　）$_2$表示二进数，用（　）$_{10}$表示十进数．

例如：

$$\begin{aligned}(1101)_2&=(1\times8+1\times4+0\times2+1\times1)_{10}\\&=(1\times2^3+1\times2^2+0\times2^1+1\times2^0)_{10}\\&=(13)_{10}.\end{aligned}$$

表达式中各次乘幂的系数就是这个二进数的各个数位上的数码．同样，

$$(111001)_2=(1\times2^5+1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0)_{10}$$
$$=(57)_{10};$$

$$(0.1011)_2=\left(1\times\frac{1}{2}+0\times\frac{1}{4}+1\times\frac{1}{8}+1\times\frac{1}{16}\right)_{10}$$
$$=(1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4})_{10}$$
$$=(0.6875)_{10};$$

$$(110.101)_2=(1\times2^2+1\times2^1+0\times2^0$$
$$+1\times2^{-1}+0\times2^{-2}+1\times2^{-3})_{10}$$
$$=\left(6+\frac{1}{2}+\frac{1}{8}\right)_{10}$$
$$=(6.625)_{10}.$$

十进制记数法的基数是“十”；二进制记数法的基数是“二”．无论是一个十进数还是二进数，它的各个数位上的数码，就是将此数写成以相应进位制的基数为底数的乘幂的和式时，和式中各项的系数．反过来，已知此和各项的系数，这个数在相应的进位制中各个数位上的数码就有了，并排写出这些数码就得到这个数．对于其他进位制，也是这样．

在电子计算机中采用的是二进数，但二进数数位多，写起来很不方便，而且只有两个不同的数码，容易产生错误．为了解决这个问题，在编制电子计算机解题程序时，常用八进数（见本书第三部分）．这样一方面可以避免上述二进数的缺点，另一方面因八进数与二进数之间有着简单的关系，互相转换也很方便（见下节）．

1.3 八进制与十六进制

当我们用“逢八进一”的方法计数时，得到八进制记数法．

试用"逢八进一"的方法来数图1中的火柴的根数；怎样用八进数将所得结果表示出来呢？

在八进制记数法中，有八个不同的数码：0、1、2、3、4、5、6、7，"逢八进一"．八进制记数法的基数是"八"．相应的位置法则是：小数点前第一位（自右至左）上一个单位是"一"，第二位上一个单位是"八"，第三位上一个单位是"六十四"等等；小数点后第一位（自左至右）上一个单位是"八分之一"，第二位上一个单位是"六十四分之一"，第三位上一个单位是"五百一十二分之一"等等．

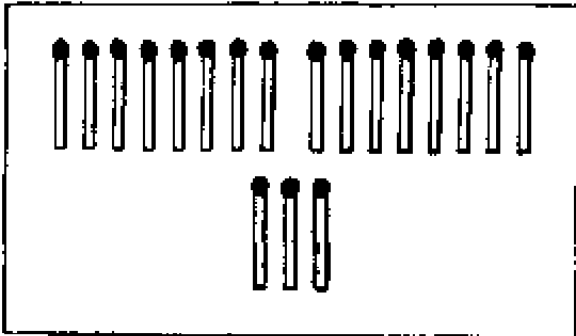


图 6

我们数火柴时，先将八根放在一堆，得到二堆，还剩下三根（图6）．于是，火柴根数用八进数表示时就为$(23)_8$．

根据"逢八进一"与位置法则，我们可写出下列对照表：

十进数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…
八进数	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	…

八进数	0.1	0.01	0.001	0.0001	…
十进数	$\frac{1}{8}$	$\frac{1}{8^2}$	$\frac{1}{8^3}$	$\frac{1}{8^4}$	…

任意一个八进数可写作基数"八"的乘幂的和的形式．

例如：

$$(17)_8=1\times 8^1+7\times 8^0=(15)_{10};$$

$$(205)_8=2\times 8^2+0\times 8^1+5\times 8^0=(133)_{10};$$

$$\begin{aligned}(0.21)_8&=2\times 8^{-1}+1\times 8^{-2}=\frac{2}{8}+\frac{1}{64}=\frac{17}{64}\\&=(0.265\cdots)_{10};\end{aligned}$$

$$\begin{aligned}(3412.705)_8&=3\times 8^3+4\times 8^2+1\times 8^1+2\times 8^0\\&\qquad+7\times 8^{-1}+0\times 8^{-2}+5\times 8^{-3}\\&=1802+\frac{7}{8}+\frac{5}{8^3}\\&=1802\frac{453}{512}\\&=(1802.8847\cdots)_{10}.\end{aligned}$$

学习电子计算机原理时，还要用到十六进制.

在十六进制中，有十六个不同的数码：0、1、2、3、4、5、6、7、8、9、$\bar{0}$、$\bar{1}$、$\bar{2}$、$\bar{3}$、$\bar{4}$、$\bar{5}$（在这里，记号$\bar{0}$、$\bar{1}$、$\bar{2}$、$\bar{3}$、$\bar{4}$、$\bar{5}$分别表示十、十一、十二、十三、十四、十五），“逢十六进一”，十六进制的基数是“十六”．位置法则是：小数点前第一位上（自右至左）一个单位是“一”，第二位上一个单位是“十六”，第三位上一个单位是“二百五十六”等等；小数点后第一位（自左至右）上一个单位是“十六分之一”，第二位上一个单位是“二百五十六分之一”等等.

类似于八进数，十六进数与二进数之间有简单的关系（见下节），电子计算机正是利用这种关系进行某些操作的.

上面讲了几种不同的数制，作为小结和推广，我们来看 p 进制．设 p 为任意一个大于 1 的整数，在 p 进制中，以“p”为基数，它有 p 个不同的数码：0、1、2、3、4、…、$p-1$．根据“逢 p 进一”以及如下的位置法则进行计算：小数点前第一位（自右至左）上一个单位是“一”，第二位上一个单位是“p”，第三位上一个单位是“p^2”等等；小数点后第一位（自左至右）

上一个单位是“p分之一”，第二位上一个单位是“p^2分之一”等等.

例如，若p为大于7的整数，则p进数

$$(1036.52)_p=1\times p^3+0\times p^2+3\times p+6\times p^0 +5\times p^{-1}+2\times p^{-2}.$$

注意，等式右端是十进数. 根据数的乘幂的规定：

$$p^0=1,\quad p^{-1}=\frac{1}{p},\quad p^{-2}=\frac{1}{p^2};$$

p的各次乘幂的系数为数码0、1、2、3、…、$p-1$中的某一个.

对于任意大于1的整数p，p进数

$$(a_na_{n-1}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_p$$

（其中n为非负整数，m为正整数）所表示的十进数是

$$a_n\times p^n+a_{n-1}\times p^{n-1}+\cdots+a_1\times p^1+a_0\times p^0 +a_{-1}\times p^{-1}+a_{-2}\times p^{-2}+\cdots+a_{-m}\times p^{-m}.$$

其中a_n、a_{n-1}、…、a_1、a_0、a_{-1}、a_{-2}、…、a_{-m}为数码0、1、2、3、…、$p-1$中的某一个.

2. 数制的转换

“一切矛盾着的东西，互相联系着，不但在一定条件之下共处于一个统一体中，而且在一定条件之下互相转化”. 我们可以用不同进位制的数刻划同一客观事物的数量，但这些不同进位制的数之间有着它们内在的规律性的联系. 在日常生活和工作中，我们采用十进数，而在电子计算机中则采用二进数. 要熟练地操作电子计算机，尤其是在机器算题过程中，要能迅速地处理出现的一些问题，熟悉数制间的转换是很重要的.

2.1 十进数转换成八进数

(十翻八)

我们着重讲明十进数转换成八进数的方法及原理. 至于十进数转换成其他进位制数的方法,可以同样解决.

先看怎样将十进整数转换成八进整数.

设有某种产品 461 件,现要将它们进行包装,每八件装成一盒,每八盒装成一箱. 问装成几箱,另外还有几盒零几件.

首先,每八件装成一盒,要算出 461 件产品装成的盒数及剩下的件数,只需要用 8 去除 461, 求出商数和余数:

$$
\begin{array}{r|l} 8 & 461 \\ \hline & 57 \cdots\cdots 5 \end{array}
$$

以 8 去除 461, 得商数 57, 余数 5. 即装成 57 盒还余 5 件. 再将 57 盒产品,每八盒装成一箱. 要算出箱数及余下的盒数,又只需用 8 去除 57, 求出商数和余数:

$$
\begin{array}{r|l} 8 & 57 \\ \hline & 7 \cdots\cdots 1 \end{array}
$$

以 8 去除 57, 得商数 7, 余数 1, 即装成 7 箱还余 1 盒.

总的算式可写作:

$$
\begin{array}{r|l} 8 & 461 \\ \hline 8 & 57 \cdots\cdots 5 \uparrow \\ \hline & 7 \cdots\cdots 1 \end{array}
$$

得到的结果是:装成 7 箱,另外还有 1 盒零 5 件产品.

在解决这个问题时,我们同时得出,十进整数 $(461)_{10}$ 转换成八进整数就是 $(715)_8$. 因为根据被除数、除数、商数及余数之间的关系:

被除数 = 商数 × 除数 + 余数,

在第一次所作的除法中,得到

$$461 = 57 \times 8 + 5;$$

在第二次所作的除法中，得到

$$57=7\times8+1.$$

将后式代入前式，得到

$$\begin{aligned}461&=(7\times8+1)\times8+5\\&=7\times8^2+1\times8+5\\&=7\times8^2+1\times8+5\times8^0.\end{aligned}$$

这就是说，十进整数$(461)_{10}$转换成八进整数就是$(715)_8$. 上面通过产品包装问题，我们介绍了将十进整数转换成八进整数的方法——“除八取余”法.

“要善于去观察和分析各种事物的矛盾的运动，并根据这种分析，指出解决矛盾的方法。”现在我们不用产品包装问题，而是直接论证将十进整数转换成八进整数的方法——“除八取余”法.

已知十进整数$(746)_{10}$，要求将它转换成八进整数.

在前面我们考察数字表达原理时看到，要将一个十进数转换成八进数，只需将此数写成八的各次乘幂的和的形式，然后将各次乘幂的系数并排写出，就是所求的八进数. 对于十进整数$(746)_{10}$来说，如果有

$$\begin{aligned}(746)_{10}=a_n\times8^n+a_{n-1}\times8^{n-1}+\cdots+a_2\times8^2\\+a_1\times8^1+a_0\times8^0,\end{aligned}\qquad(1)$$

其中n为非负整数，a_n、a_{n-1}、…、a_2、a_1、a_0为八进制数码0、1、2、…、7中的某一个. 则所求的八进整数就是

$$(a_na_{n-1}\cdots a_2a_1a_0)_8,$$

即

$$(746)_{10}=(a_na_{n-1}\cdots a_2a_1a_0)_8.$$

问题在于怎样确定数n以及数a_n、a_{n-1}、…、a_2、a_1、a_0. 我们观察等式(1)，等式右端前n项中皆含因数8，最末一项

则不含因数8，提出此前n项的公因数8，(1)式便可改写作：

$$(746)_{10}=(a_n\times8^{n-1}+a_{n-1}\times8^{n-2}+\cdots+a_2\times8+a_1)\times8+a_0;$$

等式右端括号中的数是整数，且a_0为数码0、1、2、…、7中的某一个．这就是说，用8去除十进整数746，得到商数

$$q_1=a_n\times8^{n-1}+a_{n-1}\times8^{n-2}+\cdots+a_2\times8+a_1,\qquad(2)$$

而余数则为a_0.

我们来看等式(2)的右端，类似于等式(1)，我们又可将等式(2)改写作：

$$q_1=(a_n\times8^{n-2}+a_{n-1}\times8^{n-3}+\cdots+a_2)\times8+a_1;$$

同样，右端括号中的数是整数，且a_1为数码0、1、2、…、7中的某一个．这就是说，用8去除商数q_1，得到商数

$$q_2=a_n\times8^{n-2}+a_{n-1}\times8^{n-3}+\cdots+a_2,$$

而余数则为a_1.

再用8去除商数q_2，得到的余数即为a_2；继续用这个方法，直到商数为0时，余数就是最高一位(即最左面的一位)上的八进制数码a_n.

这样，我们就得到了数n及数码a_n、a_{n-1}、…、a_2、a_1、a_0.

根据上面分析的结果来做，就是

```
8| 7 4 6
 8| 9 3……2   (以8去除746，得商数q1=93，余数a0=2；)
 8| 1 1……5   (以8去除93，得商数q2=11，余数a1=5；)
  8| 1……3    (以8去除11，得商数q3=1，余数a2=3；)
     0……1    (以8去除1，得商数q4=0，余数a3=1。)
```

由于商数已为零，故所得的余数即为最高位的系数．因而我们得到

$$(746)_{10}=(1352)_8.$$

这就是"除八取余"法.

练习 将十进整数 2695 转换成八进整数.

答案: $(2695)_{10}=(5207)_8$.

下面再看怎样将十进小数转换成八进小数.

我们先看纯小数(即整数部分为零)的情况.

"对于具体的事物作具体的分析。""不同质的矛盾,只有用不同质的方法才能解决。"将十进纯小数转换成八进纯小数的方法不再是"除八取余"法了,而是"乘八取整"法. 下面,我们来观察分析一个具体的例子.

已知十进纯小数 $(0.4736)_{10}$, 要求将它转换成八进纯小数. 要解决这个问题, 也是要通过将此十进纯小数写成八的各次乘幂(此时为负整数幂)的和的形式. 如果有

$$\begin{aligned}(0.4736)_{10}&=a_1\times 8^{-1}+a_2\times 8^{-2}+a_3\times 8^{-3}+\cdots\\&\quad+a_{m-1}\times 8^{-(m-1)}+a_m\times 8^{-m}+\cdots\\&=\frac{a_1}{8}+\frac{a_2}{8^2}+\frac{a_3}{8^3}+\cdots+\frac{a_{m-1}}{8^{m-1}}+\frac{a_m}{8^m}+\cdots;\end{aligned}\quad(3)$$

其中 m 为正整数, a_1、a_2、a_3、…、a_{m-1}、a_m、…为八进制数码 0、1、2、3、…、7 中的某一个. 则得

$$(0.4736)_{10}=(0.a_1a_2a_3\cdots a_{m-1}a_m\cdots)_8.$$

我们来看由已知十进纯小数, 如何确定等式 (3) 中各个系数 a_1、a_2、a_3、…、a_{m-1}、a_m、….

等式(3)右端各项的分母都是八的乘幂,而第一项的分母为八的一次幂,其他各项的分母中,幂指数都高于一. 用 8 乘十进纯小数 0.4736, 得到

$$(0.4736)_{10}\times 8=a_1+a_2\times 8^{-1}+a_3\times 8^{-2}+\cdots$$
$$+a_{m-1}\times 8^{-(m-2)}+a_m\times 8^{-(m-1)}+\cdots$$
$$=a_1+\left(\frac{a_2}{8}+\frac{a_3}{8^2}+\cdots+\frac{a_{m-1}}{8^{m-2}}+\frac{a_m}{8^{m-1}}+\cdots\right).$$

因为 a_2、a_3、…、a_m、…是数码 0、1、2、…、7 中的某一个，故等式右端括号中是纯小数，乘积$(0.4736)_{10}\times 8$ 的整数部分即为 a_1，小数部分则为

$$b_1=(0.4736)_{10}\times 8-a_1$$
$$=a_2\times 8^{-1}+a_3\times 8^{-2}+\cdots$$
$$+a_{m-1}\times 8^{-(m-2)}+a_m\times 8^{-(m-1)}+\cdots$$
$$=\frac{a_2}{8}+\frac{a_3}{8^2}+\cdots+\frac{a_{m-1}}{8^{m-2}}+\frac{a_m}{8^{m-1}}+\cdots;$$

再用 8 乘此小数部分，得到

$$8b_1=a_2+a_3\times 8^{-1}+\cdots$$
$$+a_{m-1}\times 8^{-(m-3)}+a_m\times 8^{-(m-2)}+\cdots$$
$$=a_2+\left(\frac{a_3}{8}+\cdots+\frac{a_{m-1}}{8^{m-3}}+\frac{a_m}{8^{m-2}}+\cdots\right);$$

此时，所得乘积的整数部分为 a_2，小数部分为

$$b_2=8b_1-a_2$$
$$=a_3\times 8^{-1}+\cdots+a_{m-1}\times 8^{-(m-3)}+a_m\times 8^{-(m-2)}+\cdots$$
$$=\frac{a_3}{8}+\cdots+\frac{a_{m-1}}{8^{m-3}}+\frac{a_m}{8^{m-2}}+\cdots;$$

继续用 8 乘 b_2，所得整数部分即为 a_3，小数部分则为

$$b_3=8b_2-a_3;$$

…………………………………………

一般来说，我们只要求出若干位小数，所以上述步骤作过若干次以后即停止.

根据以上观察和分析，我们来求 a_1、a_2、…、a_{m-1}、a_m.

计算	说明
0.4736	
× 8	
3.7888	(以 8 去乘 0.4736, 得整数部分为 $a_1=3$, 小数部分为 $b_1=0.7888$;)
× 8	
6.3104	(以 8 去乘 0.7888, 得整数部分为 $a_2=6$, 小数部分为 $b_2=0.3104$;)
× 8	
2.4832	(以 8 去乘 0.3104, 得整数部分为 $a_3=2$, 小数部分为 $b_3=0.4832$;)
× 8	
3.8656	(以 8 去乘 0.4832, 得整数部分为 $a_4=3$, 小数部分为 $b_4=0.8656$;)
× 8	
6.9248	(以 8 去乘 0.8656, 得整数部分为 $a_5=6$, 小数部分为 $b_5=0.9248$.)
⋮	

一般来说，有限位的十进纯小数转化为八进纯小数后，不一定是有限位的. 因此，$(0.4736)_{10}=(0.36236\cdots)_8$. 如果我们要求取四位有效数字，根据在八进制里三舍四入的原则，得到

$$(0.4736)_{10}\approx(0.3624)_8.$$

这就是"乘八取整"的方法. 要注意，用 8 作乘法时，只乘小数部分；第一次所得整数部分是所求八进纯小数的小数点后第一位(自左至右)上的数码；将每次所得的整数顺次一位一位排下去就得到所求八进纯小数. 对于任意一个十进纯小数，皆可用"乘八取整"的方法转换成八进纯小数.

练习 将十进纯小数 0.634 转换成八进纯小数(取三位有效数字).

答案：$(0.634)_{10}\approx(0.505)_8$。

用"除八取余"的方法可将十进整数转换成八进整数，用"乘八取整"的方法可将十进纯小数转换成八进纯小数. 在这个基础上，我们可将任意一个已知十进小数转换成八进小数。为此，我们只要将整数部分按"除八取余"法得到所求八进小数的整数部分；再将纯小数部分按"乘八取整"法得到所求八

进小数的纯小数部分．整数部分与纯小数部分合起来就是所求的八进小数．

例如，根据前面已有的结果：

$$(746)_{10}=(1352)_8;$$

$$(0.634)_{10}=(0.505)_8;$$

便可得到

$$(746.634)_{10}=(1352.505)_8.$$

练习　将十进小数 1728.346 转换成八进小数．

答案：$(1728.346)_{10}=(3300.261)_8$.

2.2 十进数转换成二进数

（十翻二）

“矛盾的普遍性即寓于矛盾的特殊性之中。” 明了了将十进数转换成八进数的方法的实质，就容易得出将十进数转换成二进数的方法了．

对于十进整数 746，用“除二取余”的方法可化为二进整数．由

```
2 | 746        余数
2 | 373 ······ 0   ↑
2 | 186 ······ 1   |
2 |  93 ······ 0   |
2 |  46 ······ 1   |
2 |  23 ······ 0   |
2 |  11 ······ 1   |
2 |   5 ······ 1   |
2 |   2 ······ 1   |
2 |   1 ······ 0   |
      0        1   |
```

得到：

$$(746)_{10}=(1011101010)_2.$$

对于十进纯小数 0.634，用"乘二取整"的方法可化为二进纯小数．由

$$
\begin{array}{r l}
0.634 & \\
\times\quad\quad 2 & \text{整数部分} \\
\hline
1.268 & 1 \\
\times\quad\quad 2 & \\
\hline
0.536 & 0 \\
\times\quad\quad 2 & \\
\hline
1.072 & 1 \\
\times\quad\quad 2 & \\
\hline
0.144 & 0 \\
\times\quad\quad 2 & \\
\hline
0.288 & 0 \\
\times\quad\quad 2 & \\
\hline
0.576 & 0 \\
\times\quad\quad 2 & \\
\hline
1.152 & 1 \\
\times\quad\quad 2 & \\
\hline
0.304 & 0 \\
\times\quad\quad 2 & \\
\hline
0.608 & 0 \\
\times\quad\quad 2 & \\
\hline
1.216 & 1 \downarrow
\end{array}
$$

得到：

$$(0.634)_{10}=(0.1010001001\cdots)_2\approx(0.101000101)_2.$$

于是，十进数$(746.634)_{10}$转换成二进数就有

$$(746.634)_{10}\approx(1011101010.101000101)_2.$$

2.3 八进数与二进数间的相互转换

在前节中，我们已经提到，用二进制写出一个数，位数太多．由于八进数与二进数之间有着特别简单的关系，所以八进数与二进数之间的相互转换十分方便．因此，人们在书写时常常用八进数．现在我们来看八进数与二进数相互转换的方法．

对于八进数与二进数的相互转换方法，我们注意下面两点：

第一，八进制中有八个不同的数码 0、1、2、3、4、5、6、7，将每一个数码用三位二进制数表示，便得到：

八进数	0	1	2	3	4	5	6	7
二进数	000	001	010	011	100	101	110	111

第二，数 8 与数 2 之间有如下的乘幂关系：

$$8=2^3.$$

我们来看将八进整数转换成二进整数的情形．已知八进整数$(527)_8$，要求将它转换成二进整数．

因为 $$(527)_8=5\times 8^2+2\times 8^1+7\times 8^0,$$

以关系式 $8=2^3$ 代入，得到

$$\begin{aligned}(527)_8&=5\times(2^3)^2+2\times(2^3)^1+7\times(2^3)^0\\&=5\times 2^6+2\times 2^3+7\times 2^0.\end{aligned}\tag{4}$$

在二进制中，只有两个不同的数码 0、1．由于上式右端 2 的乘幂项的系数大于 1，不是二进制数码，我们还不能立刻写出所要求的二进整数．为此，将系数 5、2、7 写作三项“二”的乘幂的和：

$$5=4+1=2^2+1=1\times 2^2+0\times 2^1+1\times 2^0;$$

$$2=2^1=0\times 2^2+1\times 2^1+0\times 2^0;$$

$$7=4+2+1=2^2+2^1+2^0=1\times 2^2+1\times 2^1+1\times 2^0.$$

将所得的结果代入(4)式的右端，得到

$$\begin{aligned}(527)_8&=(1\times 2^2+0\times 2^1+1\times 2^0)\times 2^6+(0\times 2^2+1\times 2^1\\&\quad+0\times 2^0)\times 2^3+(1\times 2^2+1\times 2^1+1\times 2^0)\times 2^0\\&=1\times 2^8+0\times 2^7+1\times 2^6+0\times 2^5+1\times 2^4+0\times 2^3\\&\quad+1\times 2^2+1\times 2^1+1\times 2^0.\end{aligned}$$

等式右端为 2 的各次乘幂的和，且系数为 1 或 0（为二进制数码）．由这个表达式就可将八进整数 $(527)_8$ 写成二进整数．

所以

$$(527)_8=(101010111)_2.$$

观察上面所得的结果，二进整数 101010111 中，三组数码 101、010、111 分别是八进数码 5、2、7 写成的三位二进数．由此得出，将八进整数转换为二进整数的一般方法：已知一个八进整数，我们将它的每一个数码写成三位二进数，依次并排写起来，就将这个八进整数转换成二进整数了．

例如，要将八进整数 $(1352)_8$ 转换成二进整数，只要将这个八进整数的每一个数码写成三位二进数：

八进数	1	3	5	2
	↕	↕	↕	↕
二进数	001	011	101	010

于是得到

$$(1352)_8=(1011101010)_2.$$

与通常写出整数的方法一样，最高数位上数码 0 略去不写，但中间及末尾数位上的 0 不可略去。

用同样方法可以将八进小数转换成二进小数．例如，将八进纯小数 0.505 转换成二进纯小数．将小数点后各个八进制数码分别写成三位二进数：

八进数	5	0	5
	↕	↕	↕
二进数	101	000	101

于是得到

$$(0.505)_8=(0.101000101)_2.$$

对于八进小数 $(1352.505)_8$，要将它转换成二进小数，我们只要将各个整数位上的八进制数码及各个小数位上的八进

制数码分别写成三位二进数，依次排列起来即得

$$(1352.505)_8=(1011101010.101000101)_2.$$

从上面所讲到的将八进数转换成二进数的方法中，同时也就得到将二进数转换成八进数的方法．给出一个二进数，我们将小数点前(自右至左)每三个数码合写成一个八进数码，再将小数点后(自左至右)每三个数码合写成一个八进数码，依次排列起来就得到所求的八进数．

例如，将下列二进数转换成八进数：

(1) $(101111010100)_2$.

由

二进数	101	111	010	100
	↕	↕	↕	↕
八进数	5	7	2	4

得到

$$(101111010100)_2=(5724)_8.$$

(2) $(1010011.01111)_2$.

由

二进数	001	010	011	.	011	110
	↕	↕	↕		↕	↕
八进数	1	2	3	.	3	6

得到

$$(1010011.01111)_2=(123.36)_8.$$

注意：在最高整数位（小数点前最左面一位）前面添 0 或在小数点后最末位(最右面一位)后添 0，不改变此数．

练习 将二进数$(11101.10111001)_2$转换成八进数．

答案：$(11101.10111001)_2=(35.562)_8$.

最后指出，当我们用"除二取余"以及"乘二取整"的方法

将十进数转换成二进数时，操作步骤较长，因此我们常将十进数先转换成八进数（“除八取余”，“乘八取整”步骤较少），然后再将所得的八进数转换成二进数.

例如，将十进数$(746.634)_{10}$转换成二进数. 根据前面的已知结果：

$$(746.634)_{10}=(1352.505)_8;$$

$$(1352.505)_8=(1011101010.101000101)_2;$$

所以

$$(746.634)_{10}=(1011101010.101000101)_2.$$

2.4 十六进数与二进数间的相互转换

类似于八进制与二进制之间的关系，十六进制与二进制之间同样有着十分简单的关系.

在十六进制中，有十六个不同的数码. 将它们写成四位二进数，得到

十六进数	0	1	2	3	4	5	6	7
	↕	↕	↕	↕	↕	↕	↕	↕
二 进 数	0000	0001	0010	0011	0100	0101	0110	0111
十六进数	8	9	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$
	↕	↕	↕	↕	↕	↕	↕	↕
二 进 数	1000	1001	1010	1011	1100	1101	1110	1111

根据$16=2^4$，容易得出将十六进数转换成二进数的方法. 例如，将十六进整数$(18\bar{5})_{16}$转换成二进整数，只要将十六进数码1、8、$\bar{5}$分别写成四位二进数码：

$$\begin{array}{ccc} 1 & 8 & \bar{5} \\ \updownarrow & \updownarrow & \updownarrow \\ 0001 & 1000 & 1111 \end{array}$$

然后按次序排列起来，就得到所求的二进数了.

$$(18\bar{5})_{16}=(110001111)_2.$$

关于将十六进小数转换成二进小数，以及将二进小数转

换成十六进小数的方法，读者不难得出．这里我们只列举几个例子：

(1) 将十六进小数$(2037.594)_{16}$转换成二进小数．

$(2037.594)_{16}=(10000011010111.010110010100)_2$.

(2) 将二进数$(10110101110)_2$转换成十六进数．

$(10110101110)_2=(5\bar{0}\bar{4})_{16}$.

(3) 将二进小数$(1100100101.101001)_2$转换成十六进小数．

$(1100100101.101001)_2=(325.\bar{0}4)_{16}$.

以上我们讲到十进数转换为八进数，十进数转换为二进数以及八进数、十六进数与二进数之间的相互转换的方法．那么，非十进数，如何转换为十进数呢？

2.5 八进数转换为十进数

（八翻十）

我们来看一个例子，将八进整数$(4372)_8$转换成十进整数．先将八进数$(4372)_8$写成“八”的各次乘幂的和的形式：

$$(4372)_8=4\times8^3+3\times8^2+7\times8^1+2\times8^0.$$

然后，计算等式右端表达式的值，就得出所求十进数．当然，我们可以直接计算出各项的值，然后再求和；然而为了计算方便，还可以将这个和式改写成如下累次相乘、相加的形式：

$$\begin{aligned}(4372)_8&=4\times8^3+3\times8^2+7\times8^1+2\times8^0\\&=[(4\times8+3)\times8+7]\times8+2.\end{aligned}$$

为了计算等式右端表达式的值，我们写出八进数$(4372)_8$的各个数码，并将基数8写在右上角，再按下列步骤计算：

	4	3	7	2	8
+)		32	280	2296	
	4	35	287	2298……所求的十进数．	

(1) 将第一个数码 4 写在横线的下面.

(2) 将横线下数码 4 乘以 8，写在第二个数码 3 的下面；求出和数 3+32=35，写在横线的下面.

(3) 将横线下数码 35 乘以 8，写在第三个数码 7 的下面；求出和数 7+280=287，写在横线的下面.

(4) 将横线下数码 287 乘以 8，写在第四个数码 2 的下面，求出和数 2+2296=2298.

最后得到的数字就是所求的十进数，即

$$(4372)_8=(2298)_{10}.$$

一般而言，已知一个八进整数

$$(x)_8=(a_n a_{n-1} a_{n-2}\cdots a_1 a_0)_8,$$

由

$$\begin{aligned}&(a_n a_{n-1} a_{n-2}\cdots a_1 a_0)_8\\&=a_n\times 8^n+a_{n-1}\times 8^{n-1}+a_{n-2}\times 8^{n-2}+\cdots+a_1\times 8+a_0\\&=(\cdots((a_n\times 8+a_{n-1})\times 8+a_{n-2})\times 8+\cdots+a_1)\times 8+a_0\end{aligned}$$

依次求出：

$$a_n\times 8+a_{n-1}=q_1;$$
$$q_1\times 8+a_{n-2}=q_2;$$
$$q_2\times 8+a_{n-3}=q_3;$$
$$\cdots\cdots\cdots\cdots$$
$$q_{n-2}\times 8+a_1=q_{n-1};$$
$$q_{n-1}\times 8+a_0=(x)_{10}.$$

$(x)_{10}$ 就是所要求的结果.

具体计算时，可以写成下列格式：

$$\begin{array}{lcccccc|l} & a_n & a_{n-1} & a_{n-2}\cdots & a_1 & a_0 & & 8 \\ +) & & 8a_n & 8q_1\ \cdots & 8q_{n-2} & 8q_{n-1} & & \\ \hline & a_n & q_1 & q_2\ \cdots & q_{n-1} & (x)_{10} & & \end{array}$$

$(x)_{10}$……所求的十进数.

对于八进小数来说，我们只需把它适当扩大一个倍数，化为八进整数，然后再根据上面所说的方法化为十进整数，最后将所得的结果缩小同样的倍数，就得到所求的十进小数了.

例如，将八进数$(0.4372)_8$化为十进数.

$$
\begin{aligned}
(0.4372)_8 &= 4\times 8^{-1}+3\times 8^{-2}+7\times 8^{-3}+2\times 8^{-4} \\
&= 8^{-4}\times(4\times 8^3+3\times 8^2+7\times 8+2) \\
&= 8^{-4}\times 2298 \\
&= \left(\frac{2298}{4096}\right)_{10} \\
&\approx (0.5610)_{10}.
\end{aligned}
$$

上面我们讲了几种常用的数制及其相互转换的问题，作为小结．我们指出，对于几种常用的数制间的相互转换，应该掌握以下几个基本方法：

(1) 十翻八.

整数部分用“除八取余”法，纯小数部分用“乘八取整”法，即可实现转换.

(2) 八翻十.

将八进数写成八的乘幂的和的形式，计算出它的值，即可实现转换.

(3) 八翻二，二翻八.

利用下表：

八进数	0	1	2	3	4	5	6	7
二进数	000	001	010	011	100	101	110	111

将八进数的每个数码用对应的三位二进数码写出，即可实现转换.

(4) 十六翻二，二翻十六.

利用下表：

十六进数	0	1	2	3	4	5	6	7
二进数	0000	0001	0010	0011	0100	0101	0110	0111
十六进数	8	9	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$
二进数	1000	1001	1010	1011	1100	1101	1110	1111

将十六进数的每个数码用对应的四位二进数码写出，即可实现转换.

这四种常用数制间的其他转换，都可以用上面的几个本方法来实现.

例如，要十翻二，可以先做十翻八，再做八翻二. 我们不在此一一列举了.

三、二进制与电子计算机

我们已经知道，数的概念和数制的产生都是人类长期生产实践的结果，并且从理论上讨论了几种常用数制及其相互转换的方法．同时，我们还看到了同一个数可以有多种不同的表示形式．例如，数"八"在十进制中用$(8)_{10}$表示，在八进制中用$(10)_8$表示，在二进制中用$(1000)_2$表示．尽管形式不同，但是它们所反映的事物的数量是完全相同的

"**理论的基础是实践，又转过来为实践服务。**"我们研究不同数制的目的就是为了适应实践中各种不同的需要，更好地解决实际问题．在本节中，我们将着重讨论二进制与电子计算机之间的一些关系．例如，为什么大多数电子计算机是采用二进制的？生产实践问题中的数据大多数是十进制的，那么这些数是怎样输入到电子计算机中去的？电子计算机又是怎样对十进数进行二进数的运算的？以及利用电子计算机来演算数学问题的步骤是怎样的等等

在回答这些问题之前，我们先来看一个例子．请想一下，利用算盘计算

$$x=(30+15-18)\times 10\div 5$$

的过程是怎样的（见下页表）？

在电子计算机中，基本上也是按照类似的步骤进行计算的．这里算盘就相当于电子计算机中的运算器和寄存器，纸就相当于计算机中的存储器和寄存器．因为整个计算过程是由人来控制的，所以人相当于起了控制器的作用．

我们把利用算盘进行计算的全过程列成以下的表格：

次序	操作	数字	说明
1	取数	30	把 30 拨到算盘上.
2	+	15	在算盘上做加法 30+15，并把结果 45 留在算盘上.
3	−	18	在算盘上做减法 45−18，并把结果 27 留在算盘上.
4	×	10	在算盘上做乘法 27×10，并把结果 270 留在算盘上.
5	÷	5	在算盘上做除法 270÷5，并把结果 54 留在算盘上.
6	送数	x	把结果 $x=54$ 写在纸上.
7	停止		

从表格中看出，除了 30、15、18、10、5 这些数字外，还有 +、−、×、÷、取数、送数、停止等操作符号. 这些符号在电子计算机中也是用数字表示的，称为操作码. 表格中每一横行里包括次序、操作码、被操作的数字内容，在电子计算机中称为一条指令. 不过在电子计算机中，被操作的数字内容，并不是写数字本身，而是写存储器中存放这个数字的房间号码，也称为地址码. 无论是操作码、地址码和指令在计算机中都是用人们事先规定的数字来表达的（这一方面的内容在后面还要具体介绍）.

由一串指令和数组成的整个计算步骤就称为一个程序. 从上面的说明可以看出，整个计算程序完全可以用数字表达出来，然后以一定的方式输入机器，经过运算就可以得到所需要的结果.

因此，制造和使用电子计算机都必须和数字打交道，究竟采用怎样的数制表示的数字最方便呢？下面就来讨论这个问题.

1. 为什么大多数电子计算机都采用二进制

由于人们平时习惯了用十进制进行计算，所以最早的计算工具都是采用十进制的．例如算盘、手摇计算机、电动计算机等．随着电子工业的发展，近代出现了用电子元件组成电子线路的电子计算机，使计算的速度得以成万倍地提高．在计算机的设计和制造的实践中，人们发现用电子线路来表达二进数要比表达十进数具有更多的优点，因此，现在大多数电子计算机中都采用二进制的数字电路．

在电子计算机中采用二进制究竟有哪些优点呢？这可以从二进制的一些特点谈起．

1.1 表示二进数的物理方法容易实现

在计算机中，进行运算的数码以及操作码、地址码、指令，如果都是用二进数来表示的话，由于在二进数的任何一个数位上出现的数码不是“0”就是“1”，因此，要在机器中用物理方法表示一位二进数，只需要一种具有两个稳定状态的元件就可以了．

现在已经找到许多种元件具有两个稳定状态．例如，继电器触点的闭合和断开；晶体管的饱和与截止；磁芯的两种不同的剩磁状态等等．这些都可以用来方便地表示一位二进数．

下面，我们就来简单地介绍一下利用磁芯的两种不同的剩磁状态表示一位二进数的情况．

磁芯是用一种特殊性质的磁性材料做成的．现在常用的是一种叫做铁氧体（或称作铁淦氧）的磁性材料．把这种材料压制成环状（见图 1）就构成了磁芯，大概只有半粒芝麻那样大小．

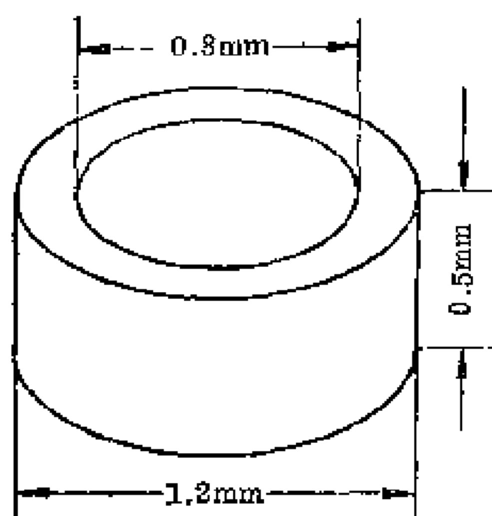


图 1

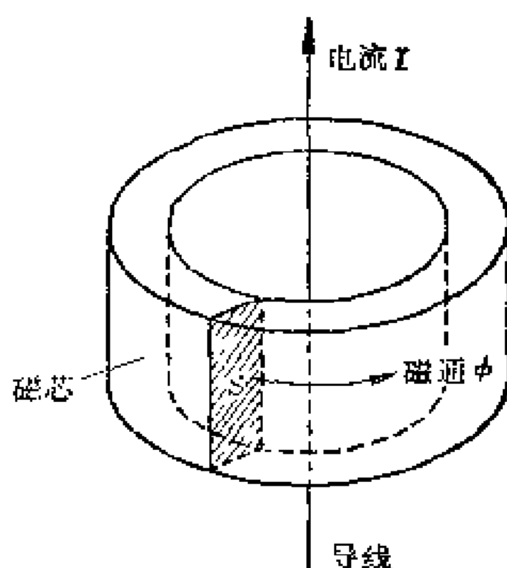


图 2

将一根导线穿过磁芯，并在导线里通以电流 I，磁芯就会磁化产生磁通 ϕ（见图 2）．通俗一点讲，就是通电以后产生了一个磁场，磁通 ϕ 就是通过面积 S 中的磁力线的条数．当电流 I 足够大，达到 I_m 时，磁化达到饱和状态，这时磁通为 ϕ_m；如果电流 I 消失，在磁芯中还有剩磁 ϕ_r．反之当电流足够大达到 $-I_m$ 时，磁芯就会反向磁化，并达到另一个饱和点 $-\phi_m$；此时若电流再消失，磁芯具有另一种剩磁 $-\phi_r$．

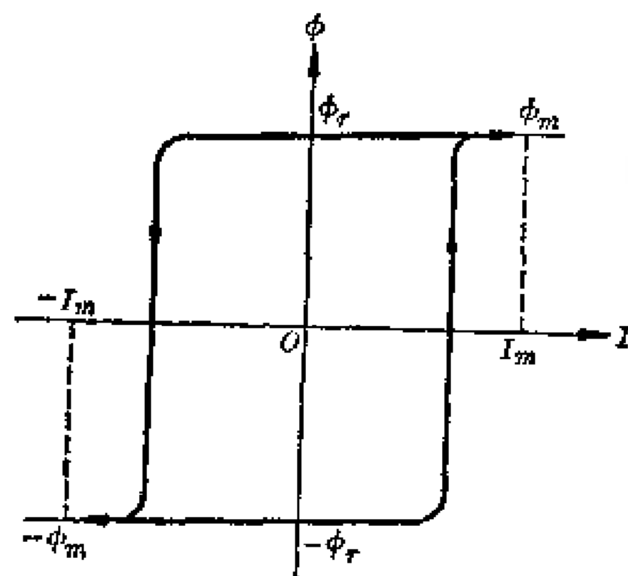


图 3

电流 I 和磁通 ϕ 之间的变化规律可以由图 3 表示出来．磁芯的这两种不同的剩磁状态 ϕ_r 和 $-\phi_r$，就可以用来表示一位二进数的两个数码“1”和“0”．

此外，在电子计算机中还用脉冲的有无，脉冲的极性（正、

负），电位的高低（0 伏和 −6 伏）等简单而可靠的方法来进行数的存放和传送，这里就不多谈了.

1.2 采用二进制可以节省计算机的存储设备量

我们知道任何一个十进数和二进数（不考虑数的符号）分别可以写成下列形式：

$$a_n\times10^n+a_{n-1}\times10^{n-1}+\cdots+a_1\times10+a_0\times10^0$$
$$+a_{-1}\times10^{-1}+\cdots+a_{-m}\times10^{-m},$$

其中 a_n、a_{n-1}、…、a_1、a_0、…、a_{-m} 为数码 0、1、2、3、4、5、6、7、8、9 中的某一个；

$$b_n\times2^n+b_{n-1}\times2^{n-1}+\cdots+b_1\times2+b_0\times2^0$$
$$+b_{-1}\times2^{-1}+\cdots+b_{-m}\times2^{-m},$$

其中 b_n、b_{n-1}、…、b_1、b_0、…、b_{-m} 为数码 0、1 中的某一个.

因此，用十进制表示零、一、二、三、四、五、六、七、八、九这十个数，需要 0、1、2、3、4、5、6、7、8、9 这十个数码组成的一位数.

如果把数码与位数的乘积称为设备状态．那么，这时共有十个设备状态（图 4）.

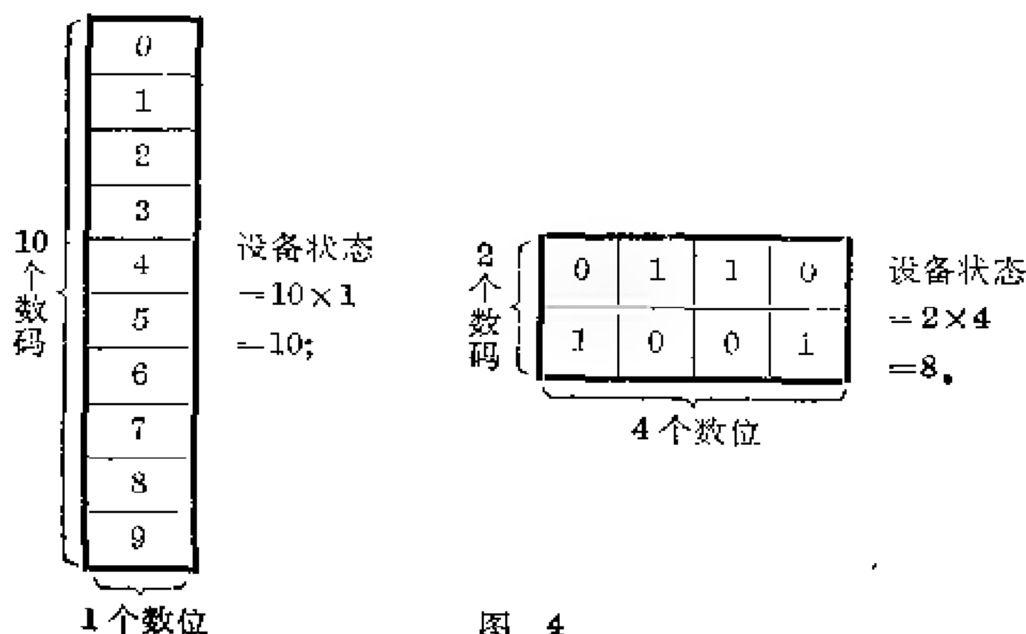


图 4

因为这十个数中最大的数

$$(9)_{10}=(1001)_2,$$

所以用二进制表示同样这十个数，需要0、1这两个数码组成的四位数，因此只有八个设备状态．并且$(1111)_2=(15)_{10}$，说明四位二进数还不止表示到十进数九，实际上可以表示到十进数十五．

如果计算机的存储设备量与设备状态成正比的话，那么在机器中采用二进制比采用十进制能节省设备量．严格地说，存储设备仅是计算机的一部分，要使机器完整有用，还必须要有许多附加设备．所以存储设备量与设备状态成正比的假定仅仅是粗略的．

我们再来讨论用十进数表示零到九十九这一百个数需要多少设备状态．因为这时需要0～9这十个数码组成的二位数，因此共有二十个设备状态．而这一百个数中最大的数$(99)_{10}=(1100011)_2$，所以用二进数表示这一百个数需要0、1这二个数码组成的七位数．因此，共有十四个设备状态．并且$(1111111)_2=(127)_{10}$，所以实际上七位二进数可以表示到十进数一百二十七．

利用

$$\lg 2=0.3010,$$

可以得到

$$2^N=10^{N\lg 2}\approx 10^{0.3N}.$$

这个式子表示一个N位二进数相当于$0.3N$位十进数．

从上面的讨论知道，表示一个N位二进数需要$2\times N=2N$个设备状态，而表示一个$0.3N$位十进数需要$10\times 0.3N=3N$个设备状态．所以一般来说，在电子计算机中采用二进制要比十进制节省存储设备量．

练习 计算一下，用十进数和二进数表示零到九百九十九这一千个数分别需要多少设备状态？

答案：30，20．

采用二进制能比十进制节省存储设备量，是否还有比二进制更节省设备量的数制呢？下面就来简单地讨论一下这个问题．

假设在机器中最节省设备量的数制是 x 进制，又设 N 是用 x 进制所表示的数的位数，则 $x^N-1=M$ 就是表示 N 位的 x 进数中最大的数．例如：

$10^2-1=99$ 就是表示二位十进数中最大的数是 $(99)_{10}$．

$2^2-1=3$ 就是表示二位二进数中最大的数是 $(3)_{10}=(11)_2$．

如果设 N 位 x 进数所需的设备状态为 y，那末 $y=x\times N$，再利用 $x^N-1=M$，可得

$$\ln(M+1)=N\times\ln x$$

$$=\frac{y}{x}\times\ln x.$$

所以

$$y=\ln(M+1)\cdot\frac{x}{\ln x}.$$

上式表明，当所表示的数 M 一定时，所需要的设备量 y 是进位制的基数 x 的函数．现在我们的问题是要求出最节省设备的数制 x 进制，因此可以把问题归结为求 y 的极小值．

因为当 M 确定时，$\ln(M+1)$ 是常数，所以求 y 的极小值又可归结为求比 $\frac{x}{\ln x}$ 的极小值．x 的取值范围是不等于 1 的正数，但是取小于 1 的正数 x 作为数制的基数是不适宜的，故下面只考虑 $x>1$ 的情况，列出比值表：

x	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	$e=2.71828\cdots$
$\frac{x}{\ln x}$	6.58	4.16	3.40	3.06	2.86	2.79	2.74	2.72	$e=2.71828\cdots$
x	2.8	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5
$\frac{x}{\ln x}$	2.719	2.73	2.79	2.89	2.99	3.11	3.23	3.35	3.47
x	7.0	7.5	8.0	8.5	9.0	9.5	10	100	…
$\frac{x}{\ln x}$	3.60	3.72	3.85	3.97	4.10	4.22	4.34	21.71	…

可以看出，当 $x=e\approx 2.71828$ 时，比值

$$\frac{x}{\ln x}=\frac{e}{\ln e}=e\approx 2.71828$$

达到极小值．因而 y 有极小值 $e\times\ln(M+1)$.

如果大家学过微积分的话，那末只要将

$$y=\ln(M+1)\cdot\frac{x}{\ln x}$$

对 x 求导数，并且从导数 $\frac{dy}{dx}=0$ 中，求出极小值即可．因为从

$$\frac{dy}{dx}=\ln(M+1)\left[\frac{\ln x-\frac{1}{x}\cdot x}{(\ln x)^2}\right]$$

$$=\ln(M+1)\left[\frac{1}{\ln x}-\frac{1}{(\ln x)^2}\right]=0,$$

即从

$$\frac{\ln(M+1)}{(\ln x)^2}(\ln x-1)=0$$

中可以得出：当 $\ln x-1=0$，即 $x=e$ 时，y 有极小值

$$e\cdot\ln(M+1).$$

因此，e 进制的设备量最小．但基数是非整数的数制是不适宜的；而与 e 最接近的整数是三，所以用三进制最节省设备．与 e 接近的另一个整数是二，因此除了三进制外，二进制比其他数制更节省设备．由于二进制的其他一些特点（例如，上面谈到的表示二进数的物理方法比较容易实现），所以大多数电子计算机还是普遍采用二进制．

1.3 二进数的运算十分简便

在二进制中，只有两个不同的数码"0"和"1"，它有如下的加法和乘法：

$$0+0=0, \qquad 0\times 0=0,$$
$$0+1=1, \qquad 0\times 1=0,$$
$$1+0=1, \qquad 1\times 0=0,$$
$$1+1=10; \qquad 1\times 1=1.$$

列成加法表和乘法表是：

+	0	1
0	0	1
1	1	10

×	0	1
0	0	0
1	0	1

这个规则和十进数的运算规则类似（不同的是 $1+1=10$），但比十进制的加法表和乘法表要简单得多．

我们知道，在十进制里的算术运算是根据"逢十进一"、"借一当十"的法则进行的．而在二进制里，我们只要根据"逢二进一"、"借一当二"的法则，就可以对二进数进行算术运算了．

下面举出几个二进数算术运算的例子，大家可以从中领会二进数的运算方法．

加法

$$\begin{array}{r} 1101.001 \\ +)\quad 110.01 \\ \hline 10011.011 \end{array}$$

减法

$$\begin{array}{r} 1101.001 \\ -)\quad 110.01 \\ \hline 110.111 \end{array}$$

乘法

$$\begin{array}{r} 1101.001 \\ \times)\quad 110.01 \\ \hline 1101001 \\ 1101001 \\ 1101001 \\ \hline 1010010.00001 \end{array}$$

除法

$$\begin{array}{r} 10.0001 \\ 11001\,\overline{)\,110100.1} \\ 11001 \\ \hline 101000 \\ 11001 \\ \hline 1111 \end{array}$$

这里，我们利用

$$1101.001 \div 110.01 = 110100.1 \div 11001$$

来进行计算.

在电子计算机中采用二进数，不仅可以使机器结构简单，而且还可以用某些方法来提高运算速度，特别是乘法的速度．例如乘数中出现“1”时，可以将被乘数不变地写在相应的位置上，又可以将乘数中接连的几个“0”跳过不用。

练习 计算　$11011.11+100.11$；

$11011.11-100.11$；

11011.11×100.11；

$11011.11\div100.11$。

在数字电子计算机中，除了加、减、乘、除算术运算以外，还有一种是逻辑运算，下面我们先来简单地介绍一下逻辑加和逻辑乘.

逻辑加用符号“∨”表示．逻辑加法表如下：

∨	0	1
0	0	1
1	1	1

对二进数来说：

$$0 \vee 0=0;\ 0 \vee 1=1;\ 1 \vee 0=1;\ 1 \vee 1=1.$$

它表示参加逻辑加运算的两个数 A“或”B 中只要有一个是 1 时和是 1，否则和是 0．可以看出逻辑加和普通加很相像，除了 $1 \vee 1=1$ 外，其他三种情况的结果是一样的．因此，逻辑加是按位运算的，没有进位．例如：

逻辑加

$$\begin{array}{r} 1101 \\ \vee)\ 1111 \\ \hline 1111 \end{array}$$

普通加

$$\begin{array}{r} 1101 \\ +)\ 1111 \\ \hline 11100 \end{array}$$

逻辑乘用符号“∧”表示．逻辑乘法表如下：

∧	0	1
0	0	0
1	0	1

对二进数来说：

$$0 \wedge 0=0;\ 0 \wedge 1=0;\ 1 \wedge 0=0;\ 1 \wedge 1=1.$$

它表示参加逻辑乘的两个数 A“与”B 都是 1 时积是 1，否则积是 0．逻辑乘运算规则从表面上看和普通乘法一样，但逻辑乘与逻辑加一样，也是按位运算的（不斜角乘）．例如：

逻辑乘

$$\begin{array}{r} 1101 \\ \wedge)\ 1111 \\ \hline 1101 \end{array}$$

普通乘

$$\begin{array}{r} 1101 \\ \times)\ 1111 \\ \hline 1101 \\ 1101 \\ 1101 \\ 1101 \\ \hline 11000011 \end{array}$$

还有一个逻辑运算称为逻辑非，在参加运算的数上面画一划来表示.

对二进数来说，

$$\bar{0}=1,\ \bar{1}=0.$$

它表示参加逻辑非运算的数 A 若是 0，则 $\bar{A}$ 是 1；反过来，若 A 是 1，则 $\bar{A}$ 是 0.

对于这三种逻辑运算，在计算机内有三种基本的逻辑线路："或"门、"与"门和"非"门．通常称为门电路.

"或"门的作用是完成逻辑加运算．用逻辑框图表示如下：

$A \longrightarrow$ [或] $\longrightarrow C=A\vee B.$
$B \longrightarrow$

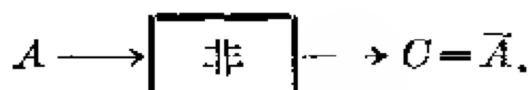
当输入线 A"或"B 中只要一条有信号时，输出线 C 就出现信号.

"与"门的作用是完成逻辑乘运算．用逻辑框图表示如下：

$A \longrightarrow$ [与] $\longrightarrow C=A\wedge B.$
$B \longrightarrow$

当输入线 A"与"B 中同时有信号时，输出线 C 才有信号出现；如果 A"与"B 中任一个没有信号，则输出线 C 也就没有信号.

“非”门的作用是完成逻辑非运算．用逻辑框图表示如下：



当输入线 A 有信号时，输出线 C 没有信号；当输入线 A 没有信号时，输出线 C 有信号．

如果规定出现信号表示“1”，没有信号表示“0”，那末“或”门、“与”门、“非”门就能完成二进数的逻辑加、逻辑乘和逻辑非三种运算了．

因为二进数的算术运算可以通过逻辑运算来实现，所以在电子计算机中，利用各种不同作用的门电路构成的逻辑线路，可实现二进数的算术运算以及其他一些操作．

下面，我们就以两个二进数 X 和 Y 相加为例来说明这一点．

考察数 X 和 Y 的第 i 位上的数码 x_i (0 或 1) 和 y_i (0 或 1) 相加的情况．这时，除了要考虑 x_i 和 y_i 的值，还要考虑由低一位（第 $i-1$ 位）上来的进位数 c_{i-1} (0 或 1)，相加的结果得到一个和 s_i 及一个进位数 c_i．这个加法规则包括以下八种可能的情况：

x_i	0	1	0	0	1	1	0	1
y_i	0	0	1	0	1	0	1	1
c_{i-1}	0	0	0	1	0	1	1	1
s_i	0	1	1	1	0	0	0	1
c_i	0	0	0	0	1	1	1	1

因此，二进数 X 和 Y 的每一位 x_i 和 y_i 的加法都可以用下面由“与”门、“或”门、“非”门组成的逻辑线路来实现：

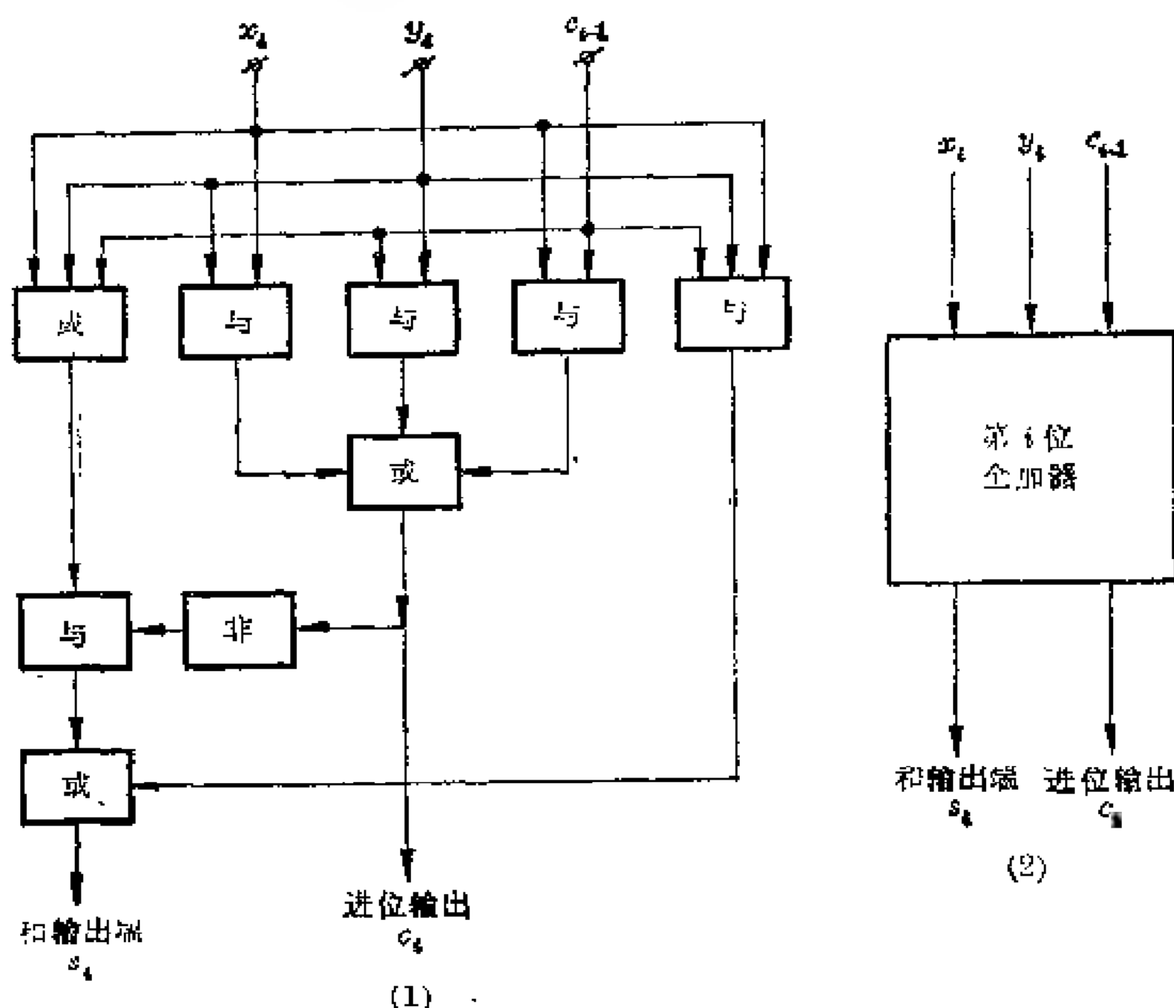


图 5

图 5 (1) 表示的这个逻辑线路是由“与”、“或”和“非”三种门电路构成的，一般就称为全加器，可简记作图 5 (2). 它可以完成二进数 X 和 Y 相加时，其中某一位的相加和进位，就是当 x_i、y_i、c_{i-1} 为上面表中八种情况之一时，相应的和 s_i 和进位 c_i 也是上面八种情况之一.

例如，当 $x_i=1$、$y_i=0$、$c_{i-1}=1$ 时，必有 $s_i=0$、$c_i=1$. 因为根据逻辑线路可以写出下列式子(在下一节中将进一步说明为什么可以这样写)：

$$进位\ c_i=(x_i\wedge y_i)\vee(y_i\wedge c_{i-1})\vee(x_i\wedge c_{i-1}).$$

为求出 c_i 的值，可将 $x_i=1$、$y_i=0$、$c_{i-1}=1$ 代入上式，并利用逻辑加和逻辑乘的运算法则，得到

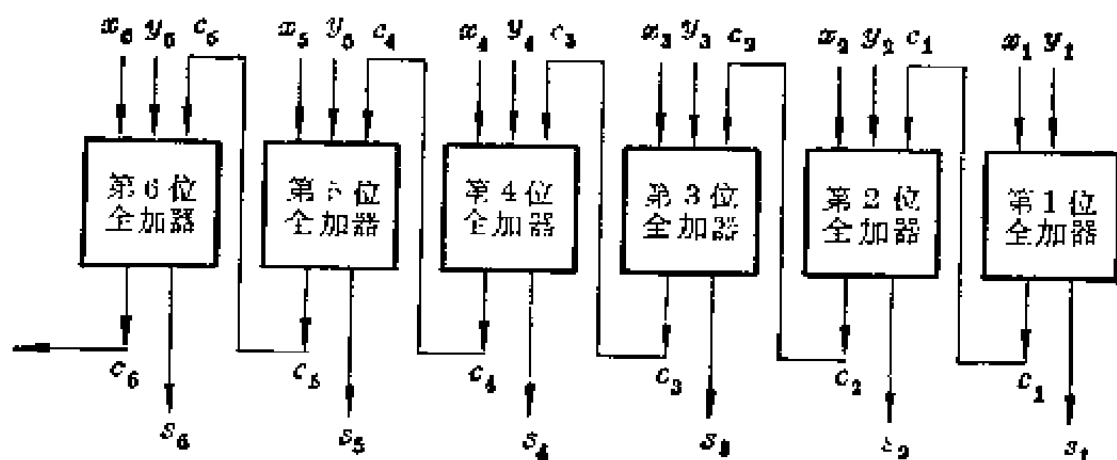
$$\begin{aligned}进位\ c_i&=(1\wedge 0)\vee(0\wedge 1)\vee(1\wedge 1)\\&=0\vee 0\vee 1\\&=0\vee 1\\&=1.\end{aligned}$$

同样，

$$\begin{aligned}和\ s_i&=(x_i\vee y_i\vee c_{i-1})\\&\quad\wedge\overline{(x_i\wedge y_i\vee y_i\wedge c_{i-1}\vee x_i\wedge c_{i-1})}\vee(x_i\wedge y_i\wedge c_{i-1})\\&=(1\vee 0\vee 1)\wedge\bar{1}\vee(1\wedge 0\wedge 1)\\&=1\wedge 0\vee 0\\&=0.\end{aligned}$$

对于其他七种情况，同样可以验证成立。

如果二进数 X 和 Y 各有 N 位，那么把 N 个全加器恰当地连接起来，就构成了一个 N 位二进数的加法器，下面是一个加法器的逻辑框图：



一位二进数的加法，可由门电路组成的逻辑线路全加器来实现．而 N 位二进数的加法，是由 N 个全加器构成的逻辑线路来实现的．因此，我们看到了二进数的加法运算确实是

可以通过"或"、"与"、"非"三种逻辑运算来实现. 其他算术运算如何通过逻辑运算来实现的问题就不在此详述了. 并且可以指出, 单独使用"与"门、"或"门是很少的, 而常用的是"与非"门、"与或非"门等复合电路的形式. 它们的基本原理是相同的.

1.4 采用二进数就能用逻辑代数作为工具研究逻辑线路

首先简单介绍一下什么是逻辑代数和逻辑线路. 在实际生活中就存在着最简单的逻辑代数和逻辑线路. 我们先举几个例子来说明.

[例 1] 某试验需一盏电灯 P, 为了使用方便, 同时装有两个开关 A 和 B. 这两个开关连接成这样, 当任一开关合上时, 电灯就接通电流而发亮; 也就是说, 当某一开关"或"另一开关合上时电灯就发亮. 自然, 若两个开关同时合上时, 电灯也会发亮.

拨"开"与拨"关"是两种状态, 在数学上可以用"1"和"0"来代表"开"和"关". 同理, "亮"和"不亮"这两种状态也可以用"1"和"0"来代表. 于是 A、B、P 的状态之间的关系, 可以用 A、B、P 取两个不同的值"0"和"1"的形式来表示:

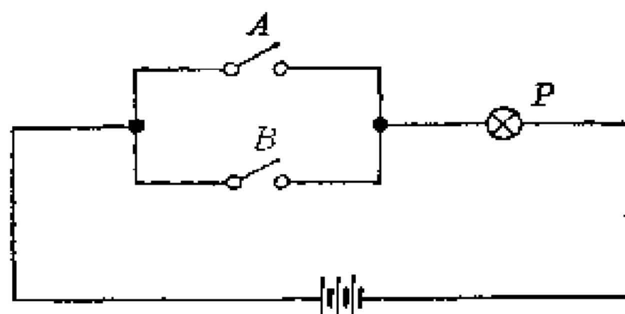
		B	
	P	0	1
A	0	0	1
	1	1	1

从这个表中可以看出, 这就是上节所讲的逻辑加运算:

$$A \vee B = P.$$

上式是一个最简单的逻辑代数式. 实现这个逻辑运算的线路

称为逻辑线路．$A \vee B = P$ 的逻辑线路就是上节提到的“或”门电路．这个问题代表的“或”门电路图如下：



［例 2］ 车间的电灯 P，当且仅当厂里的电源总闸刀 A “与”车间里的电源闸刀 B 都合上后才能发亮．

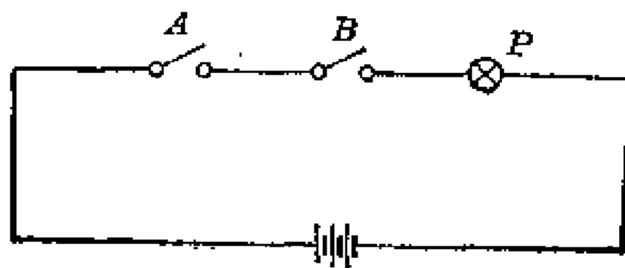
“不合”与“合”上闸刀是两种状态，我们用“0”和“1”来代表；“无电”和“有电”两种状态也分别用“0”和“1”来代表．于是 A、B、P 的状态之间的关系也可以用 A、B、P 取两个不同的值“0”和“1”的形式来表示：

		B	
	P	0	1
A	0	0	0
	1	0	1

从这个表中可以看出，这就是上节所讲的逻辑乘运算：

$$A \wedge B = P.$$

其相应的逻辑线路是“与”门电路，这个问题的“与”门电路图如下：



[例 3] 为了安全生产，车间里的电源闸刀 A 合上时，串联在电路中的指示红灯 P 就亮了，表示已经通电，当闸刀离开时，指示灯就不亮。

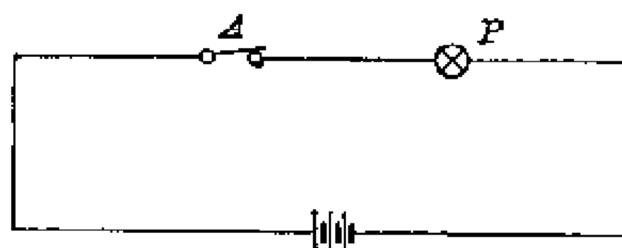
如果用"0"和"1"表示"不合"和"合"这两种不同的状态，用"0"和"1"表示"亮"与"不亮"这两种不同的状态．那末 A、P 之间的状态关系，可以用 A、P 取两个不同的值"0"和"1"的形式表示：

A	0	1
P	1	0

从这个表中可以看出，这就是上节所讲的逻辑非运算：

$$\overline{A}=P.$$

其相应的逻辑线路是"非"门电路，这个问题的"非"门电路图如下：



从这些简单的例子中可以看到，逻辑代数就是对仅取两种不同值"1"和"0"的变量间的逻辑运算求值的数学．而实现这些逻辑代数式的线路就叫做逻辑线路．由于在逻辑代数的运算中变量和结果都是只能取两个值"0"和"1"，它们分别代表实际问题里的两种不同状态，人们往往把这种变量理解为一个"开关"，于是逻辑代数也常常被称为开关代数．逻辑线路常常被称为开关线路．

人们在很早就从实践中总结了逻辑代数的数学原理，后

来发展成为数理逻辑学中重要的数学工具．近代由于生产的不断发展，逻辑代数的应用已深入到许多部门．特别是逻辑代数被应用到电路设计中以后，它有力地推动了电子计算机和各种自动控制系统的发展．

下面，我们来简单地介绍一下逻辑代数在电子计算机中的应用．

如果用符号表示参加运算的数或未知量(当然它们的值只可能是 0 和 1 这两个数)，那末

$$C=A\vee B$$

表示当 A“或” B 有一个是 1 时，C 为 1；否则 C 为 0．

$$C=A\wedge B$$

表示仅当 A“与” B 都是 1 时，C 才为 1；否则 C 为 0．

$$C=\overline{A}$$

表示当 A 是 0 时，则 C 是 1；当 A 是 1 时，则 C 是 0．

根据上述规定，可以得出下列基本关系：

$$A\vee 0=A;\quad A\vee 1=1;\quad A\vee A=A;$$

$$A\wedge 0=0;\quad A\wedge 1=A;\quad A\wedge A=A;$$

$$A\vee\overline{A}=1;\quad A\wedge\overline{A}=0;\quad \overline{\overline{A}}=A;$$

$$\overline{A\vee B}=\overline{A}\wedge B;\quad \overline{A\wedge B}=\overline{A}\vee B.$$

普通代数中的运算定律——交换律、结合律和分配律在逻辑代数中也适用，即

$$A\vee B=B\vee A;\quad (A\vee B)\vee C=A\vee(B\vee C);$$

$$A\wedge B=B\wedge A;\quad (A\wedge B)\wedge C=A\wedge(B\wedge C);$$

$$A\wedge B\vee A\wedge C=A\wedge(B\vee C);$$

$$A\vee B\wedge C=(A\vee B)\wedge(A\vee C).$$

以上关系可由逻辑加、逻辑乘、逻辑非运算的意义直接验证，例如，验证分配律

$$A \wedge B \vee A \wedge C = A \wedge (B \vee C)$$

成立，可以列出表格：

A	B	C	$A\wedge B$	$A\wedge C$	$A\wedge B\vee A\wedge C$	$B\vee C$	$A\wedge(B\vee C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1

对 A、B、C 取 0 和 1 有八种可能情况，都得出

$$A\wedge B\vee A\wedge C \quad 与 \quad A\wedge(B\vee C)$$

的作用是一样的．因此，分配律

$$A\wedge B\vee A\wedge C = A\wedge(B\vee C)$$

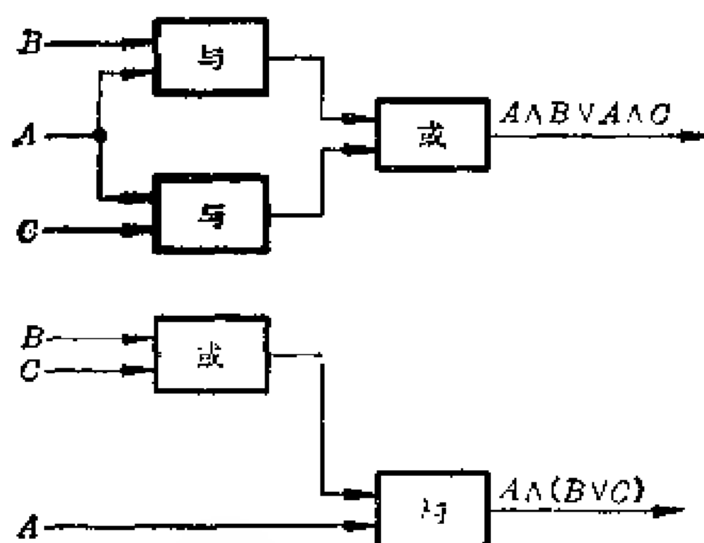
成立．

上面已经讲过，计算机中各种算术运算可以用门电路组成各种不同作用的逻辑线路来实现．逻辑线路能够输入信号和输出信号，这种信号都是双值的，即在一条信号线上或者有信号出现或者没有信号出现，此时分别代表 1 或 0 这两个值．并且逻辑线路中输入信号和输出信号之间存在着预定的函数关系．

例如，实现逻辑运算

$$A\wedge B\vee A\wedge C \quad 以及 \quad A\wedge(B\vee C)$$

的逻辑线路的框图分别是：



上面框图中每一条信号线可能出现信号或者不出现信号．输入信号为 A、B、C，若输出信号分别为 f、g，则上面框图逻辑线路表示的逻辑函数关系分别为：

$$f=A\wedge B\vee A\wedge C;$$

$$g=A\wedge(B\vee C).$$

再以"全加器"为例，根据 x_i、y_i、c_{i-1} 三数恰有一个为 1 或者三个全是 1 时，也就是

$$x_i=1,\qquad \bar{y}_i=1,\qquad \bar{c}_{i-1}=1;$$

$$\bar{x}_i=1,\qquad y_i=1,\qquad \bar{c}_{i-1}=1;$$

$$\bar{x}_i=1,\qquad \bar{y}_i=1,\qquad c_{i-1}=1;$$

或

$$x_i=1,\qquad y_i=1,\qquad c_{i-1}=1$$

时，和 s_i 为 1，以及 x_i、y_i、c_{i-1} 中至少有二个为 1 时，即

$$x_i=1,\qquad y_i=1,\qquad \bar{c}_{i-1}=1;$$

$$x_i=1,\qquad \bar{y}_i=1,\qquad c_{i-1}=1;$$

$$\bar{x}_i=1,\qquad y_i=1,\qquad c_{i-1}=1;$$

或

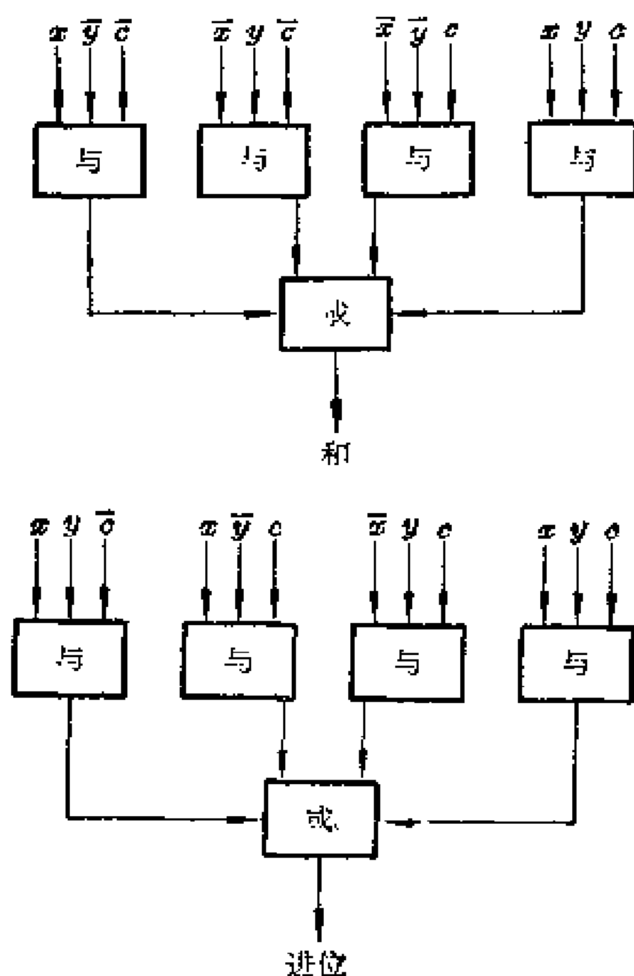
$$x_i=1,\qquad y_i=1,\qquad c_{i-1}=1$$

时，进位 c_i 为 1．于是，全加器输入信号 x_i、y_i、c_{i-1} 与输出信号 s_i、c_i 之间，分别有以下的逻辑函数关系式（省略下标 i）：

$$和=(x\wedge\bar{y}\wedge\bar{c})\vee(\bar{x}\wedge y\wedge\bar{c})\vee(\bar{x}\wedge\bar{y}\wedge c)\vee(x\wedge y\wedge c).$$

$$进位=(x\wedge y\wedge\bar{c})\vee(x\wedge\bar{y}\wedge c)\vee(\bar{x}\wedge y\wedge c)\vee(x\wedge y\wedge c).$$

这二个函数关系式所构成的全加器逻辑框图如下：



这样的逻辑线路是比较复杂的，和与进位各需要四个三端输入的“与”门和一个四端输入的“或”门．

应用逻辑代数的运算规则，就可以将上述函数关系简化，进而简化逻辑线路．

例如，可以利用

$$\bar{x}\vee x=\bar{y}\vee y=\bar{c}\vee c=1,$$

以及

$$A\vee(x\wedge y\wedge c)=A\vee(x\wedge y\wedge c)\vee(x\wedge y\wedge c)\vee(x\wedge y\wedge c),$$

简化进位线路.

$$\begin{aligned}\text{进位}&=(x\wedge y\wedge \bar{c})\vee(x\wedge \bar{y}\wedge c)\vee(\bar{x}\wedge y\wedge c)\\&\quad\vee(x\wedge y\wedge c)\vee(x\wedge y\wedge c)\vee(x\wedge y\wedge c)\\&=x\wedge y\wedge(\bar{c}\vee c)\vee x\wedge c\wedge(\bar{y}\vee y)\vee y\wedge c\wedge(\bar{x}\vee x)\\&=(x\wedge y\wedge 1)\vee(x\wedge c\wedge 1)\vee(y\wedge c\wedge 1)\\&=(x\wedge y)\vee(x\wedge c)\vee(y\wedge c).\end{aligned}$$

练习 利用逻辑代数中的基本关系和算律证明:

(1) $y\wedge(x\vee y)=y$.

(2) $\bar{x}\wedge\bar{y}\vee\bar{x}\wedge\bar{c}\vee\bar{y}\wedge\bar{c}=\overline{x\wedge y\vee x\wedge c\vee y\wedge c}$.

另一方面,如果把和理解为三个变量 x、y、c 中任何两个同时为 0, 其余一个变量为 1 时, 则和为 1; 三变量同时为 1 时, 和也为 1; 此时和的函数关系式如下:

$$\text{和}=(x\vee y\vee c)\wedge(\bar{x}\wedge\bar{y}\vee\bar{x}\wedge\bar{c}\vee\bar{y}\wedge\bar{c})\vee(x\wedge y\wedge c).$$

利用上面练习中的(2)式,可以简化和的线路.

$$\text{和}=(x\vee y\vee c)\wedge(\overline{x\wedge y\vee x\wedge c\vee y\wedge c})\vee(x\wedge y\wedge c).$$

这个函数关系式连同进位的函数关系式:

$$\text{进位}=(x\wedge y)\vee(x\wedge c)\vee(y\wedge c)$$

一起构成的逻辑线路, 就是上一节中介绍的全加器逻辑线路图. 显然那个线路比本节中未经简化的全加器逻辑线路来得简单.

以上, 我们以加法器为例简单地说明了在计算机上采用二进数后,就可以用门电路组成的逻辑线路来实现算术运算. 同时,还可以用逻辑代数作为有力的工具,对逻辑线路中输入

信号和输出信号之间存在的函数关系进行适当的变换，达到简化线路的目的。

2. 怎样把十进数输入到电子计算机中去

2.1 二进制编码的十进数 （十——二进制）

由于二进数有以上四个特点，所以大多数电子计算机都采用二进数．但是我们平时接触到的数字大都是十进数，所以我们如果利用电子计算机算题时，必须首先把原始数据从十进制转换成二进制，经过电子计算机计算，最后还要把所得的结果从二进制转换成十进制．数制间的转换工作量是很大的，为了减少人工操作，就要设法把数制间的转换工作由机器来完成，这就涉及到怎样把十进数输入到机器中去的问题．

下面，我们先来看一个类似的问题．大家知道，当甲和乙两人要利用电报通讯时，甲必须把电文中的每一个汉字先请邮局的同志按照电码译成相应的4个十进制数码，然后再由发报员利用发报机以声音的长短讯号表示0、1、2、3、4、5、6、7、8、9这十个数码，把电码传送出去．收报员收到电码后，又要把电码译成相应的汉字，最后送到乙的手中．这里电码起了汉字与发报机能接受的声音讯号之间的“桥梁”作用．

十进数相当于汉字，不能直接进入机器，但是可以仿照汉字编成电码一样，把十进数编码．采用二进制编码的十进数（即十——二进制），作为十进数与电子计算机能接受的二进数之间的“中间”数制，同样发挥了“桥梁”作用．这样，就可以解决十进数输入到机器中去的问题．

将十进数用十——二进制表示的方法十分简单，我们只

要将此十进数的每一位上的数码，按下表以四位二进数写出，然后依次序排列起来，就可以得到此数的十——二进制形式了.

十进数	0	1	2	3	4	5	6	7	8	9
十——二进数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

例如，将十进数 746 写成十——二进数.

由

$$\begin{matrix} 7 & 4 & 6 \\ \updownarrow & \updownarrow & \updownarrow \\ 0111 & 0100 & 0110 \end{matrix}$$

可以写出：

$$(746)_{10}=(011101000110)_{10\ 2}.$$

练习　将下列十进数 1457, 2369, 87261 写成十——二进数.

2.2 十——二进数转换成二进数

应当注意的是，将一个十进数写成十——二进数，在形式上是象一个二进数了，因为它每一位上只出现 0、1 这两个数码. 但是实质上它并没有将十进数真正转换成二进数. 因为

$$(746)_{10}=(1352)_8=(001011101010)_2.$$

显然，这个二进数与上节中的十——二进数是不同的. 因此十——二进数的作用仅在于使十进制输入机器（具体如何输入在下节讲），它不能作为二进数参加运算.

把大量的十——二进数转换成真正的二进数的工作是由

计算机用专门的“翻译程序”自动完成的。现在就以上面的例子来看看机器中是怎样将$(746)_{10}$的十——二进数

$$(011101000110)_{10-2}$$

转换成二进数$(001011101010)_2$的。

设：

$$\alpha_1=(111100000000)_2,$$
$$\alpha_2=(000011110000)_2,$$
$$\alpha_3=(000000001111)_2,$$
$$\gamma_1=(0.00000001)_2=(2^{-8})_{10}=(16^{-2})_{10},$$
$$\gamma_2=(0.0001)_2=(2^{-4})_{10}=(16^{-1})_{10},$$
$$c_1=(001100100)_2=(10^2)_{10},$$
$$c_2=(1010)_2=(10)_{10}.$$

因为

$$(x)_{10}=(746)_{10}=a_1\times10^2+a_2\times10+a_3,$$

其中

$$a_1=7,\ a_2=4,\ a_3=6;$$

且

$$(x)_{10-2}=(011101000110)_{10-2}.$$

转换的思想方法是：先把十——二进数$(x)_{10-2}$中代表a_1、a_2、a_3的每四位二进数码

0111、0100、0110，

通过一些二进数码间的逻辑乘法和普通乘法，可以分别化为a_1、a_2、a_3这三个数码的真正二进数。

容易验证下列关系式成立：

$$a_1=((x)_{10-2}\wedge\alpha_1)\times\gamma_1,$$
$$a_2=((x)_{10-2}\wedge\alpha_2)\times\gamma_2,$$
$$a_3=(x)_{10-2}\wedge\alpha_3.$$

例如，

$$
\begin{aligned}
&((x)_{10-2}\wedge\alpha_1)\times\gamma_1\\
&=(011101000110\wedge111100000000)\times0.00000001\\
&=011100000000\times0.00000001\\
&=(000000000111)_2=(7)_{10}=a_1.
\end{aligned}
$$

因此，

$$
\begin{aligned}
(x)_{10}&=a_1\times10^2+a_2\times10+a_3\\
&=((x)_{10-2}\wedge\alpha_1)\times\gamma_1\times c_1\\
&\quad+((x)_{10-2}\wedge\alpha_2)\times\gamma_2\times c_2+((x)_{10-2}\wedge\alpha_3)\\
&=((x)_{10-2}\wedge\alpha_1)\left(\frac{10}{16}\right)^2+((x)_{10-2}\wedge\alpha_2)\left(\frac{10}{16}\right)\\
&\quad+((x)_{10-2}\wedge\alpha_3)\\
&=\left[((x)_{10-2}\wedge\alpha_1)\left(\frac{10}{16}\right)+(x)_{10-2}\wedge\alpha_2\right]\left(\frac{10}{16}\right)\\
&\quad+((x)_{10-2}\wedge\alpha_3).
\end{aligned}
$$

再设：

$$
\begin{aligned}
&c=\left(\frac{10}{16}\right)_{10}=(0.5)_8=(0.101)_2,\\
&\delta_1=(x)_{10-2}\wedge\alpha_1,\\
&\delta_2=(x)_{10-2}\wedge\alpha_2,\\
&\delta_3=(x)_{10-2}\wedge\alpha_3.
\end{aligned}
$$

则

$$(x)_{10}=(\delta_1\times c+\delta_2)\times c+\delta_3=(x)_2.$$

等式表示利用$(x)_{10}$的十——二进数$(x)_{10-2}$，通过上列一些二进制数码间的逻辑乘法、普通乘法和加法后，已得到$(x)_{10}$的真正二进数$(x)_2$了.

下面列出具体的算式：

$((x)_{10-2}=)$	011101000110
$(\alpha_1=)$	∧) 111100000000
$(\delta_1=)$	011100000000
$(c=)$	×) 0.101
	011100000000
	011100000000
$(\delta_1\times c=)$	10001100000.000
$((x)_{10-2}=)$	011101000110
$(\alpha_2=)$	∧) 000011110000
$(\delta_2=)$	000001000000
$(\delta_1\times c=)$	+) 010001100000
$(\delta_1\times c+\delta_2=)$	010010100000
$(c=)$	×) 0.101
	010010100000
	010010100000
$((\delta_1\times c+\delta_2)\times c=)$	01011100100.000
$((x)_{10-2}=)$	011101000110
$(\alpha_3=)$	∧) 000000001111
$(\delta_3=)$	000000000110
$((\delta_1\times c+\delta_2)\times c=)$	+) 001011100100
$((x)_2=(\delta_1\times c+\delta_2)\times c+\delta_3=)$	001011101010
	↓, ↓, ↓, ↓
$((x)_8=)$	1, 3, 5, 2

练习 写出$(x)_{10}=(159)_{10}$的十——二进数$(x)_{10-2}$，并利用$(x)_{10}$的十——二进数$(x)_{10-2}$得出$(x)_{10}$的真二进数$(x)_2$.

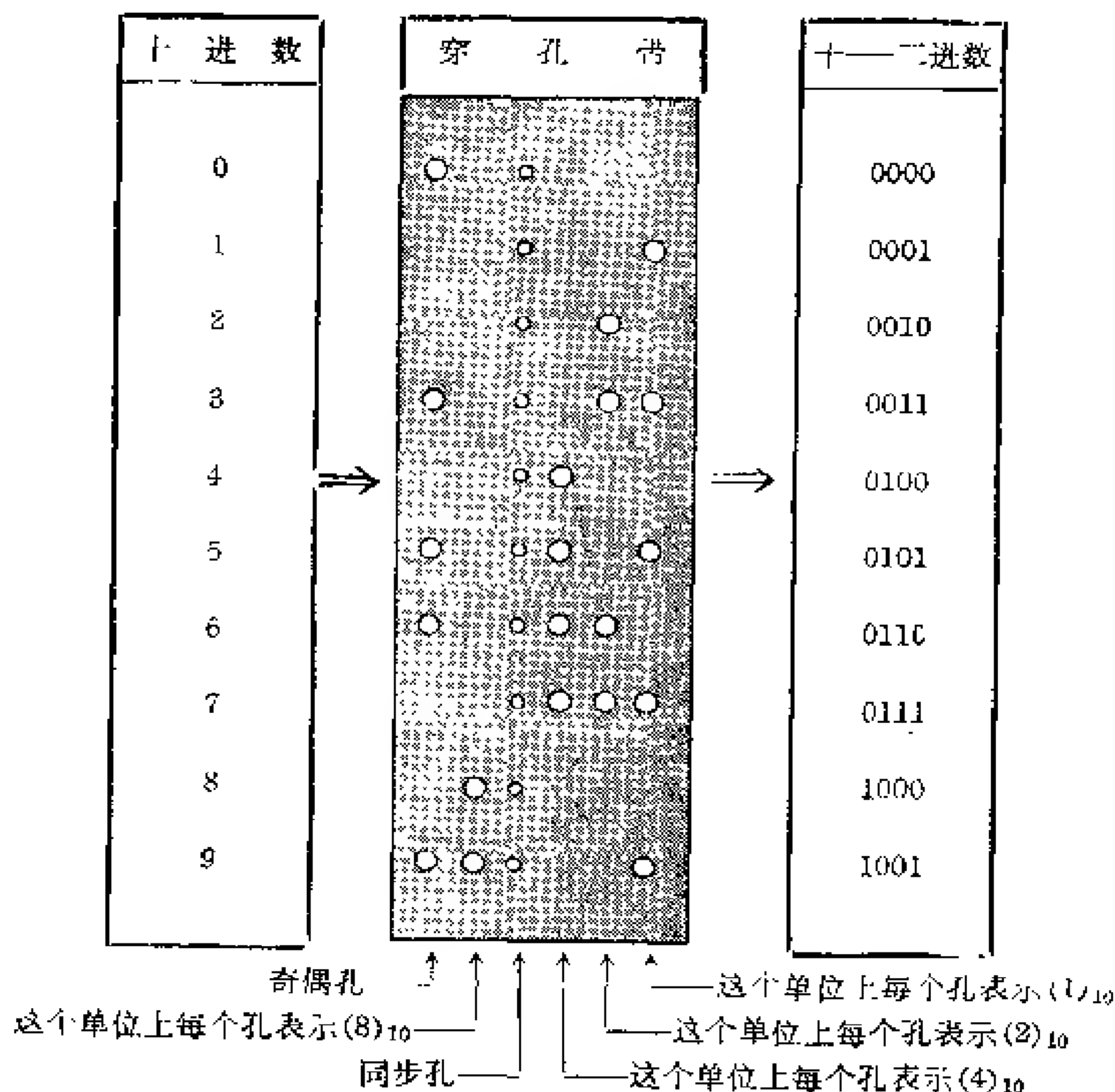
2.3 十——二进数与穿孔带

通过上面的讨论我们可以知道，在电子计算机中运算的数是二进制的，而人们经常接触的数是十进制的．为了利用电子计算机算题，我们利用二进制编码的十进数（即十——二

进制）作为中间数制，可以方便地写出每一个十进数的十——二进制形式的数，然后由机器通过专门的翻译程序自动的转换成真正的二进数。

那么每一个十——二进数又是怎样输入到机器中去的呢？常用的一个方法是利用穿孔带。所谓穿孔带就是一种纸带，它有好几种规格，其中有一种是把纸带上每一行分成五个"单位"，在一个单位上可以穿一个圆孔或不穿孔，以孔的位置和数量来表示一个十——二进数。

具体规则如下：



说明：(1)纸带中间有一列小孔(比其他五个单位上的圆孔直径小)，称同步孔，在输入数码时或拖动纸带时是有用的.

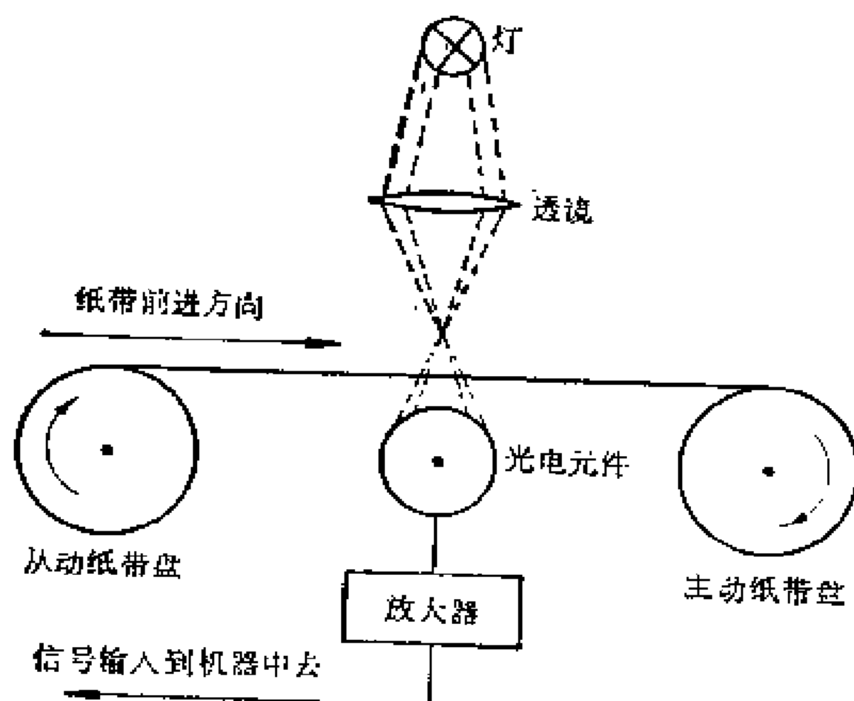
(2)纸带最左面的一列圆孔，称为奇偶孔. 由于机器校验需要，使每一行的圆孔数为奇数个.

(3) 纸带上其他各列上有圆孔的表示 1，无孔的表示 0. 如果我们不看奇偶孔和同步孔的话，那么每一行上四个单位上的圆孔就表示一个四位二进数，也就是十——二进数.

练习 读出下列纸带上的数，并分别用十进数和十——二进数写出。



利用打孔机穿孔，就能把十进数转换成十——二进数的工作，方便地通过机械操作来实现. 纸带通过光电输入机，就将十——二进数输入到机器中去. 下面是光电输入机示意图：



3. 怎样利用电子计算机来演算数学问题

在上两节中，我们已经介绍了大多数电子计算机采用二进制的原因，同时还介绍了采用二进制编码的十进数(十——二进制)，就可以将十进数输入到机器中去进行运算．在这节中，我们将通过简单的实例来概括地介绍一下利用电子计算机演算数学问题的过程．

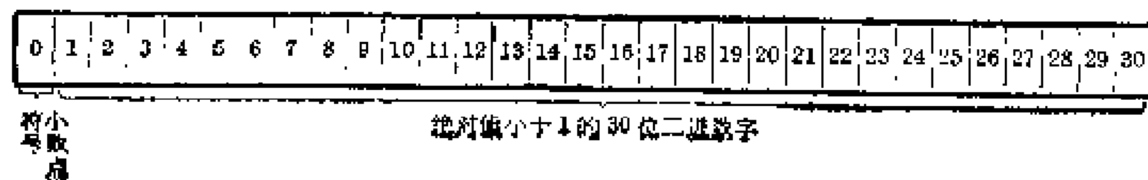
为了方便起见，我们假定使用的是国产"103"电子计算机．至于其他类型的电子计算机，它们的工作原理是类似的．

3.1 "103"电子计算机简介

"103"电子计算机是一部二地址通用数字电子计算机，它对二进数进行运算．存储器由一个磁芯体(称为快存储器)和一个磁鼓(称为慢存储器)组成．各具有1024个单元(通常称为房间)，共有2048个单元．这些房间的号码(即地址码)是以八进数来表示的，编号是从$(0000)_8$号到$(3777)_8$号．

磁芯体的每个单元是由31个磁芯串成的磁芯串．从前面知道，每一个磁芯可以表示一位二进制数码．因此，"103"机的每个存储单元可表示一个占31位的二进数，第零位为符号位，第1～30位为二进数字，利用$2^N \approx 10^{0.3N}$可以知道，它相当于九位十进数．

"103"机是一部定点机．数的表示范围为：$-1<x<1$ 也就是说，机器中的数字的小数点的位置总是固定在第零位和第一位之间(见下图)．



因此，绝对值大于或等于1的数不可能输入机器。若运算结果或中间结果大于或等于1，则发生溢出停机。为了保证机器能正常运算，在编制计算程序时，必须严格考虑好比例因子，保证初始数据、计算结果和中间结果的绝对值均小于1。

例如，在这部分最初计算的式子

$$x=(30+15-18)\times 10\div 5$$

中可以引进比例因子 $\frac{1}{60}$，将计算式子等价地化为

$$x=[(30+15-18)\times 10\div 5]\times\frac{1}{60}\times 60$$

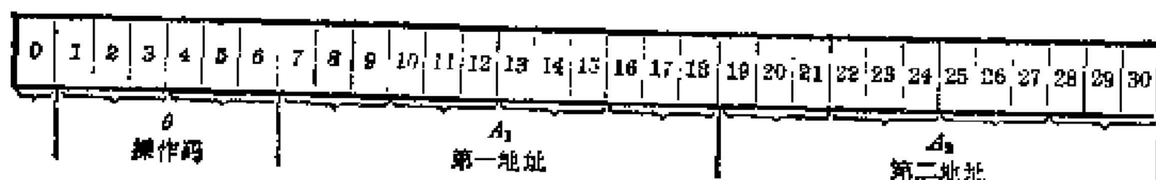
$$=\left[\left(\frac{1}{2}+\frac{1}{4}-\frac{3}{10}\right)\times\frac{1}{6}\div\frac{1}{12}\right]\times 60.$$

于是，便可以计算式子

$$x_1=\left(\frac{1}{2}+\frac{1}{4}-\frac{3}{10}\right)\times\frac{1}{6}\div\frac{1}{12};$$

最后，我们再将所得的结果 x_1 扩大60倍，就可得到所求的值 x。

"103"机可以做下列操作：加法、减法、乘法、除法、逻辑乘法、数的传送，控制转移、输入及输出、停机等。每种操作都用两个八进制数码表示，称为操作码。所谓指令就是说明从何处取数，进行何种操作，所得结果送到何处；每条指令由操作码、第一地址和第二地址构成。"103"机的指令在存储单元中的形式如下图所示：



用第零位做符号位，"0"表示正号，"1"表示负号。在定

点运算中，符号位的数码取0．第1～6位为操作码(即两个八进制数码)，第7～18位为第一地址，第19～30位为第二地址(每个地址码用四个八进制数码表示)．

例如，在存储器0001号和0002号房间中已分别存放了数$\frac{1}{2}$和$\frac{1}{4}$，取操作码$\theta=00$，第一地址$A_1=0001$，第二地址$A_2=0002$．则

$$+00\quad 0001\quad 0002$$

就组成了"103"机上的一条指令．它的具体含意是：把存储器中0001号和0002号两个房间中的数$\frac{1}{2}$和$\frac{1}{4}$取到运算器中，在运算器中做加法$\frac{1}{2}+\frac{1}{4}$；把运算结果送到寄存器B中，同时还要送到存储器的0002号房间中去．因此，0002号房间原来存放的数$\frac{1}{4}$就被改为$\frac{3}{4}$．

上面一条指令中的操作码00的作用完全是人们事先设计时规定的．在"103"机中还规定了另一些操作码．由不同的操作码构成的指令全体，称为这部机器的指令系统．为了计算前面提出的问题以及下面例题的需要，我们列出"103"机的部分指令，见表Ⅰ．

注意：若A_1、A_2地址码上带"′"则表示执行该指令时可以与这种地址码无关，故这种带"′"的地址码A_1'和A_2'在程序中可以写为任意的四个八进制数，一般写作四个0．

3.2 使用电子计算机进行演算的步骤

有了以上部分指令，我们就不难编出用"103"机计算下例的程序

［例］ 编出$x_1=\left(\frac{1}{2}+\frac{1}{4}-\frac{3}{10}\right)\times\frac{1}{6}\div\frac{1}{12}$的计算程序．

表 I

指令	操作内容	记号
$+10\ A_1\ A_2$	A_2 房间中内容 $+A_1$ 房间中内容，结果送寄存器 B 中.	$(A_2)+(A_1)\Rightarrow B$
$+20\ A_1\ A_2$	B 寄存器中内容 $+A_1$ 房间中内容，结果送寄存器 B 中，并送 A_2 房间.	$(B)+(A_1)\Rightarrow B\rightarrow A_2$
$+31\ A_1\ A_2'$	B 寄存器中内容 $-A_1$ 房间中内容，结果送寄存器 B 中.	$(B)-(A_1)\Rightarrow B$
$+41\ A_1\ A_2$	A_2 房间中内容 $-A_1$ 房间中内容，结果送寄存器 B 中，并送 A_2 房间及印出.	$(A_2)-(A_1)\Rightarrow B\Rightarrow A_2\Rightarrow Л$
$+03\ A_1\ A_2$	A_2 房间中内容 $\times A_1$ 房间中内容，结果送寄存器 B 中，并送 A_2 房间.	$(A_2)\times(A_1)\Rightarrow B\Rightarrow A_2$
$+13\ A_1\ A_2$	A_2 房间中内容 $\times A_1$ 房间中内容，结果送寄存器 B 中.	$(A_2)\times(A_1)\Rightarrow B$
$+23\ A_1\ A_2$	B 寄存器中内容 $\times A_1$ 房间中内容，结果送 B 寄存器中，并送 A_2 房间.	$(B)\times(A_1)\Rightarrow B\Rightarrow A_2$
$+33\ A_1\ A_2'$	B 寄存器中内容 $\times A_1$ 房间中内容，结果送 B 寄存器中.	$(B)\times(A_1)\Rightarrow B$
$+62\ A_1\ A_2$	B 寄存器中内容 $\div A_1$ 房间中内容，结果送 B 寄存器中，并送 A_2 房间及打印.	$(B)\div(A_1)\Rightarrow B\Rightarrow A_2\Rightarrow Л$
$+15\ A_1\ A_2$	A_1 房间中内容送 B 寄存器，并送 A_2 房间.	$(A_1)\Rightarrow B\Rightarrow A_2$
$+24\ A_1\ A_2$	B 寄存器中内容送 A_2 房间，机器转向执行 A_1 房间中的指令.	$(B)\Rightarrow A_2$、$A_1\Rightarrow I$
$+34\ A_1\ A_2$	B 寄存器中内容若大于 0，则机器转向执行 A_2 房间中的指令，否则转向执行 A_1 房间中的指令.	$\begin{cases}(B)>0 & A_2\rightarrow I\\(B)<0 & A_1\Rightarrow I\end{cases}$
$+04\ A_1'\ A_2'$	停机.	Ω

假设在存储器中存放数

$$\frac{1}{2}、\frac{1}{4}、\frac{3}{10}、\frac{1}{6}、\frac{1}{12}$$

和结果值 x_1 的房间号码暂时用记号

$$\left\langle\frac{1}{2}\right\rangle、\left\langle\frac{1}{4}\right\rangle、\left\langle\frac{3}{10}\right\rangle、\left\langle\frac{1}{6}\right\rangle、\left\langle\frac{1}{12}\right\rangle \text{ 和 } \langle x_1\rangle$$

来表示，那末上式的计算程序只需要如下五条指令：

$$\begin{array}{lll} +10 & \left\langle\frac{1}{2}\right\rangle & \left\langle\frac{1}{4}\right\rangle \\ +31 & \left\langle\frac{3}{10}\right\rangle & 0000 \\ +33 & \left\langle\frac{1}{6}\right\rangle & 0000 \\ +62 & \left\langle\frac{1}{12}\right\rangle & \langle x_1\rangle \\ +04 & 0000 & 0000 \end{array}$$

只有把程序本身(包括指令和数据)都输入到机器的存储器中后，才能使机器去执行这个程序．因此，我们把上面5条指令和5个数据依次输入到0011～0022这十个房间中去(不一定要从0011号开始)，另外规定计算的结果要送到0023这个房间中去，这样就得到了完整的程序，见表II.

表 II

程			序	说明
0011	+10	0016	0017	$\frac{1}{2}+\frac{1}{4}=\frac{3}{4}\to$ 寄存器 B.
0012	+31	0020	0000	$\frac{3}{4}-\frac{3}{10}=\frac{9}{20}\to$ 寄存器 B.
0013	+33	0021	0000	$\frac{9}{20}\times\frac{1}{6}=\frac{9}{120}\to$ 寄存器 B.

续表 11

程序				说明
0014	+62	0022	0023	$\frac{9}{120}:\frac{1}{12}=\frac{9}{10}\rightarrow$ 寄存器 $B\Rightarrow0023\rightarrow$ 打印.
0015	+04	0000	0000	停机.
0016	+40	0000	0000	"103"机中 $\frac{1}{2}$ 的八进制形式.
0017	+20	0000	0000	"103"机中 $\frac{1}{4}$ 的八进制形式.
0020	+23	1463	1463	"103"机中 $\frac{3}{10}$ 的八进制形式.
0021	+12	5252	5253	"103"机中 $\frac{1}{6}$ 的八进制形式.
0022	+05	2525	2525	"103"机中 $\frac{1}{12}$ 的八进制形式.

上述程序经过穿孔,并通过操纵台控制,经输入器就能将纸带上的程序输入到机器的 0011～0022 号房间. 然后,在操纵台上从 0011 号房间启动, 机器就会自动执行从 0011 号开始顺次下去的各条指令的要求,自动到 0016～0022 各房间中去取数和送数. 当机器正常停机后(非出错停机),在 0023 号房间中留下计算结果,并在输出机的打印纸上可以看到该结果.

在这个计算问题中的 5 个数据,因为数量较少,我们可用人工转换成八进制后直接输入. 如果有大量的十进数要输入的话,一般直接用十——二进数形式输入,再用机器中的十翻二的标准程序进行翻译工作, 这方面的内容可见专门的程序设计的书籍,在此就不详细说明了.

从上例看出,一个数学问题要在电子计算机上进行演算,大致需要做以下三项工作:

(1) 编制程序

首先分析数学公式，并根据电子计算机的指令系统，逐条编制指令，再把参加运算的数和指令统统用八进制数码来表示．这样，一个数学问题便转化为一个由一串八进制数码构成的程序．这就是所谓计算公式程序化．

(2) 纸带穿孔

其次，再把这些八进制数字代码通过穿孔机录到纸带上面，使程序由八进制数码再转化为机器能接受的一系列信息(纸带上不同位置的小孔)．这就是所谓代码信息化．

(3) 上机计算

最后把录有程序代码的纸带，通过光电输入机，把这些小孔所代表的信息转化为物理讯息，输入到机器中一系列存储单元中去(一个单元中放一条指令信息或一个数的信息)．此时，当我们启动机器开始运算时，机器即能按照已输入的程序，对给定的数据完成预定的运算．

也许有的读者要问，一个四则运算，人工只需几秒钟就可以算出来了，用电子计算机算还要编程序、穿孔、上机等一大套手续，花费这么多的时间和精力不是得不偿失吗？我们说，类似于上例的算题，用电子计算机来算是发挥不出电子计算机的优越性的．但是对于计算工作量大，重复计算部分多的计算问题，甚至依靠人工也许是不能计算得了的问题，通过编制程序后，在电子计算机上却能得到迅速的解答．例如造数学用表，在电子计算机出现以前需要几十个人工作好几年才能完成，今天在一般的电子计算机上，也只需要几个小时就能完成了．

现在，我们再来看一个略为复杂一点的例子：

设

$$y=x-\frac{x^3}{6}+\frac{x^5}{120},$$

求

$$x=k\times 0.017453 \quad (k=1,\ 2,\ \cdots,\ 45)$$

时对应的函数值 y.*

这里对于每一个 x 的值，求相应的 y 值的工作完全是重复应用公式，因此非常适合用电子计算机来进行计算.

如果程序从 0010 号开始（不一定从 0010 开始），并且规定存放自变量 x 和函数值 y 的房间号码分别是 0032、0033，存放中间结果的房间号码是 0031. 则此例的程序如下页表.

这个程序比上面一个程序长一点，我们来分析一下：指令 0011～0020 完成对每一给定的自变量 x 求出函数值 y 的工作；指令 0021、0022、0023 起到了指挥机器重复执行指令 0011～0020 的工作的作用. 程序中其他指令或数的情况比较清楚，不在此一一说明了.

从这个例子就可以看出，人们通过编制程序，能够利用电子计算机的快速特点去完成大量的重复性计算的任务.

在这一节中，我们已初步介绍了怎样编写一个数学问题的计算程序. 从中可以看到不同数制的作用. 数和指令用八进制的形式写出比较方便，但机器中运算的时候还是对二进数进行运算的. 大量的数一般用十——二进制形式输入，再用 10→2 标准程序翻译成二进数后参加运算.

上面我们介绍的这种编程序的方法是原始的编程序方法，即用机器的原始指令，由人工逐条逐条地把程序编写出

---

* 从高等数学中可以知道

$$\sin x\approx x-\frac{x^3}{6}+\frac{x^5}{120},\ \text{并且}\ 1^\circ=\frac{\pi}{180}\ \text{弧度}\approx 0.017453\ \text{弧度}.$$

所以本题实际上是计算从 1°～45° 时的正弦函数值.

表 III

程序				说明
0010	+15	0025	0032	把 0.017453 送到存放自变量 x 值的房间中去.
0011	+13	0032	0032	对每一给定的自变量 x, 求出函数值, 并打印 $x-\frac{x^3}{6}+\frac{x^5}{120}\to y\to \Pi$.
0012	+24	0013	0031	
0013	+23	0032	0033	
0014	+33	0031	0000	
0015	+33	0030	0000	
0016	+20	0032	0031	
0017	+03	0027	0033	
0020	+41	0033	0031	
0021	+00	0025	0032	修改自变量 x 的值, 即 (0032)+0.017453→0032.
0022	+31	0026	0000	当自变量 x 值未超过 45×0.017453 时, 则去求函数值 y;
0023	+34	0011	0024	当自变量 x 值超过 45×0.017453 时, 则转向停机.
0024	+04	0000	0000	停机.
0025	+01	0737	1460	$(0.017453)_{10}$ 的八进数形式.
0026	+62	2073	7562	$(46\times 0.017453)_{10}$ 的八进数形式.
0027	+12	5252	5253	$\frac{1}{6}$ 的八进数形式.
0030	+00	4210	4211	$\frac{1}{120}$ 的八进数形式.

来，通常也称为手编程序．手编程序工作是十分费时间的，它需要我们熟练地掌握机器的指令系统，这对一般使用电子计算机来解决生产实际问题的工农兵和工程技术人员来说是很不方便的．假如换了另一台不同类型的机器，由于指令系统的不同，使得一个算题的程序没有通用性．另外，手编程序从外表上来看都是八进制数码，已完全失去其数学公式的原始形式，这对其他人阅读程序来说是很困难的．当然，手编程序还有许多优点，如结构紧凑合理、计算速度快、效果高，至今仍然被不少人继续使用着．随着计算工作者编制程序的实践水平不断提高、人们在实践中就“**有所发现，有所发明，有所创造，有所前进。**”近年来已经普遍开始采用算法语言的程序来代替手编程序．只要在机器内部配有所谓“编译程序”，机器就能够根据输入的算法语言程序，自动编出相应的指令形式程序（类似于手编程序），然后，机器就执行这个编出的程序．

例如，用算法语言编制一个“已知原坐标 (x, y) 通过坐标变换公式

$$\begin{cases} x_1 = x\cos t - y\sin t + a \\ y_1 = x\sin t + y\cos t + b \end{cases}$$

求新坐标 (x_1, y_1)”的程序，可以这样书写：

```
Begin  Real a, b, t, x, y, x1, y1;
   #Read(0, '10', a, b, t, x, y);
   x1: = x* #cos(t) - y* #sin(t) + a;
   y1: = x* #sin(t) + y* #cos(t) + b;
   #Print (0, '10', x1, y1).
end 55…5 止.
    (55…5: 26个以上)
```

其中 Begin 表示程序开始，end 表示结束，Real 表示 a、b、t、x、y、x_1、y_1 都是实变量，在存储器中应为它们开好房间，以存放这些量的值. #Read(0, '10', a, b, t, x, y)表示把 a、b、t、x 和 y 都用十进数输入到机器中去. #Print(0, '10', x_1, y_1)表示把 x_1 和 y_1 都用十进数输出方法打印出来. +、−、* 表示加、减、乘的运算符号. #cos(t)、#sin(t)分别表示对应于自变量 t 的余弦值和正弦值. := 表示将其右端运算式的运算结果送入其左端变量房间中去.

为了把上述算法语言程序输入到机器中去，需要把上述程序中的每个符号按下列符号编码表中的二个八进制数码来表示.

低位 / 高位	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	·	10	a	b	c	d
2	e	f	g	h	i	j	k	l
3	m	n	o	p	q	r	s	t
4	u	v	w	x	y	z	+	−
5	*	/	÷	↓	∨	∧	¬	<
6	⩽	=	⩾	>	≠	,	:	;
7	#	'	(	[	]	'	ε	空

例如，符号 Real 在编码时应看作 ε Real ε. 故"Real a, b, t, x, y, x_1, y_1;"就编为"76 35 20 14 27 76 14 65 15 65 37

65 43 65 44 65 43 01 65 44 01 67"；又如符号"end"应编码为"76 20 31 17 76"；但结束讯号中的"55…5"在编码时是例外，应编作"55…5"而不是"0505…05"；而符号"止"由穿孔机上单独给出，不属于符号编码表范围。

因此，上述程序若全部用数码表示出来，就是：

76 15 20 22 24 31 76 76 35 20 14 27 76 14 65
15 65 37 65 43 65 44 65 43 01 65 44 01 67 70
35 20 14 17 71 00 65 75 01 00 75 65 14 65 15
65 37 65 43 65 44 72 67 43 01 66 61 43 50 70
16 32 36 71 37 72 47 44 50 70 36 24 31 71 37
72 46 14 67 44 01 66 61 43 50 70 36 24 31 71
37 72 46 44 50 70 16 32 36 71 37 72 46 15 67
70 33 35 24 31 37 71 00 65 75 01 00 75 65 43
01 65 44 01 72 76 20 31 17 76 55 55 55 55 55
55 55 55 55 55 55 55 55 止.

通过穿孔机穿孔，把上述八进制数码形式的程序代码，穿成二进制形式的程序代码，然后输入机器。所以机器所接受的算法语言程序和手编指令一样，都是二进制的代码信息。当算法语言程序进入机器后，就可以操纵存放在机内的编译程序进行工作，把输入的算法语言程序翻译为相应的指令形式的程序。所以说，机器最后正式执行的还是被翻译出来的指令形式的程序

从这个例题中，我们可以看到算法语言程序有以下几个优点：

(1) 直观性：它接近于算题中数学表达式的形式，便于使用者阅读。

(2) 通用性：它规定了统一的表达语言和符号，对不同

类型机器都可以实施.

(3) 通俗性：它用接近于人们习惯的语言结构来叙述算题过程，通俗易懂. 因此，采用算法语言后可使广大工农兵和工程技术人员，在不一定非常熟悉机器的指令系统的情况下，也可以使用电子计算机来演算题目，解决生产实际问题. 这就大大地促进了电子计算机的使用，利于普及推广. 目前，我国广大工农兵和工程技术人员在生产实践中已有越来越多的人学会了使用算法语言来编制程序. 介绍算法语言已超出了本书的范围，建议大家进一步去学习算法语言的有关书籍，同时也只有在不断地实践中才能提高编制算法语言程序的能力.